修士論文

RoboCup2D シミュレーションのアクション連鎖探索における
フィールド全体に適応した評価関数の設計と検証

Design and verification of evaluation function adapted to the entire field in action chain search of RoboCup2D simulation

報 告 者

学籍番号: 1255057
氏名: 吉見奎吾

指 導 教 員

星野孝総 准教授

令和 5 年 2 月 17 日

高知工科大学大学院工学研究科

基盤工学専攻電子・光工学コース

目 次

第1章　序論

1.1.　研究背景

近年コンピューターの性能向上に伴い，人工知能を用いた機能が注目を浴びている．特に囲碁や将棋などの対戦ゲームでは，人工知能を搭載したコンピュータが人間のプロ棋士に勝つなど，コンピュータの状況判断・決断力が人間を上回る可能性が示されている [1] [2]．これに伴って，AI の研究対象も静的環境からテレビゲームやサッカーなどの動的環境へと変化し，より実世界に近い環境での有用性を研究している．このように実世界での応用が可能となれば，人工知能がより便利で快適な世界を実現することに役立つと考えられる．

本稿では、情報ノイズや行動決定時間など現実の問題で想定される問題設定を実験環境として研究するため，サッカーシミュレーション 2D リーグ「ロボカップ 2D」（RoboCup2D）に着目した．RoboCup2D は，個別且つ特有な AI エージェントが仮想 2D フィールド上でサッカーゲームを行うゲームである。サッカーシミュレーションリーグ「RoboCup」では、「2050 年までにサッカーの世界チャンピオンチームを倒す完全自律型ヒューマノイドロボットのチームを作る」という壮大なる最終目標が設定されている [3]。また、目標達成の過程で生み出された新たな研究テーマや成果を集約し、関連技術の進歩促進も RoboCup の目的の一つとしている．RoboCup2D のプレイヤーエージェントは，0.1 秒という限られた短時間で敵味方に関係なく同時に行動決定を行う．チームのパフォーマンスは，各プレイヤーエージェントの判断力に強く依存するため，それぞれの状況に対応した柔軟な処理が必要となる．

現在、ロボカップでは多くの研究者が AI 技術を使った研究を行っている．しかし，人間のチームと戦えるようになるには時間がかかると思われる．しかし、ロボカップはサッカーというスポーツを利用しており，現代においても e スポーツやリアルなサッカーに与える影響は大きい．現在のサッカーでは，戦略解析のデジタル化が進んでおり，特に選手やボールのトラッキングの研究が目覚ましく進んでいる．実際の試合での選手の動きの詳細やパスの成功率など，さまざまなデータを IT 技術で変換・分析するシステムは，世界各国を代表チームや欧州クラブでも使っている [4,5]．一方，RobuCup では各プレーの意図や狙いを AI アルゴリズムという形で戦術もしくは戦略的知識を明示的に抽出し，それに基づいた行動決定を行うことができる．これが応用できれば，実際の相手チームの戦術や各選手の能力に近いチームを用意し，ロボカップを使って想定した戦術を検証できる上，相手チームにどれだけ有効かを確認することができる．これらを試合に反映した AI を投入することも可能である．このような試みは人間社会にとっても大きなインパクトがあり，ロボカップにとってはトップチームの戦術を取り入れてより強いチーム作るように，現代社会においても戦術的なチームを構成する時の技術プロセスになるにつながると考えられる． このように，ロボカップをフィールドとした AI 研究は，サッカーだけでなく，先にも述べたような現実世界への応用可能がある．例えば，ロボカップでは，エー

ジェントが個々のプログラムで動作し，エージェント同士のコミュニケーションによって情報を交換する．そのため，グループロボットの連携に必要な情報の受け渡しに応用することができる．また，限られた時間の中で多くの情報を処理する必要があり，実社会におけるロボットエージェントの要素を備えている．これらの部分のヒントから，自動運転や自動制御などへの応用が期待できる．

本論文では，上記の第一歩としてサッカーフィールド上でボールを運ぶ位置によって得点数やシュートできる可能性の高い位置の探索を行い，試合のなかで多くの得点を取れるチームを目指す．また同時にそれらの探索の結果から戦術を構築する際にもボールを集める最適な位置を示せるものを求めて研究を行っていく．

1.2. 研究目的

本稿では，ロボカップサッカーフィールドで得られる情報から、得点数やシュートの可能性を高めるポジションを探索し，チームの得点力向上を目指す．得点の可能性が高いポジションを視覚的に示し，現実のサッカーに応用する第一歩となることを目標に研究を進める． 秋山ら [6] は，RoboCup2D において，各サッカーエージェントの行動決定方法を木探索のアルゴリズムで実現した．この手法は，秋山らが開発したサンプルエージェント agent2d に搭載されており，2010 年に開催された大会の優勝チームである HELIOS がベースになっている．現在でも，多くのチームが agent2d をベースにしたチームを作成しており，AI 研究開発が進められている [7, 8]. この手法では，各ノードが状態，エッジが行動を表し，すべてのノードが評価関数によってスコア化される．そして，最も高い評価値を持つノードに対するアクションを次のサイクルのアクションとして選択する．評価関数の設計はプレイヤーの行動決定に大きく影響するため、本来慎重に設計する必要がある。しかし、人間がサッカーフィールドの膨大な数の状態を全て考慮して設計することは困難である．そのため，評価関数の設計は非常に難しく，選手エージェントを開発者の意図通りに動作させるためには，評価関数を詳細に設計して，それに伴う実験を繰り返し，試行錯誤を繰り返す必要がある． 本研究では，この評価関数を簡易的な $if\text{-}then$ 形式で設計する．実際は，ボールの位置座標を入力値として，得点の可能性を高める評価を出力として，最適な位置を探索する．本研究での評価関数を遺伝的アルゴリズムとファジィ推論法の一種である簡便推論法を用いて設計している．設計した評価関数の適用範囲は，限定的なフィールド範囲を想定して実験を行った．これは、それぞれの攻撃パターンの選択肢を増やし，スコアリング能力を向上させる狙いがあるためである．用いた遺伝的アルゴリズムとは、進化計算の一種である自然淘汰の仕組みを最適化問題に適用したアルゴリズムであり，高スコアが取れるようにファジィ規則の後件分のチューニングを行う．一方，用いるファジィ推論は、身の回りのあいまいな表現を扱い制御やクラス分類などができる AI 手法の一種である．このファジィ推論の中でも，計算速度の早い簡易型ファジィ推論法は、扱いやすく，遺伝的アルゴリズムを用いてチューニングが容易である．これらにより，フィールド全体でより得点の可能性が高いポジションを探索することが可能になる． 本稿の構成は以下の通りである. 第 2 章では主に RoboCup2D の概要について述べる．シミュレータの歴史，構成，構造について触れる．また，チーム作りのベースとなった agent2d について述べる．RoboCup2D では，開発を手助けするパッケージが多く用意されており，そのようなパッケージについて説明する．第 3 章では，チームの強さを判断

するための指標を決める．2020 年に開催された RoboCup2D の大会である，夏の大会と Japan Open の全試合のログファイルを収集し，そのログから得点能力が高いチームにはどうような特徴があるのかを分析する．第 4 章では，１つ目の評価関数の設計について述べる．また，１つ目の方法で評価関数を設計することとなった理由について述べる．第 5 章では，2 つ目の評価関数である遺伝的アルゴリズムとファジィ推論を用いた評価関数の設計方法について述べる．また，チューニングした評価関数がチームの強さに影響を与えるのかについて実験し考察する．第 6 章では，3 つ目の評価関数の設計と検証について述べる．また，対戦相手を変更した際の実験の有用性について述べる．最後に第 7 章では本稿についてのまとめを述べる．そして評価関数の設計やチーム作りに向けた今後の予定と，研究課題・展望について述べる．

第2章　RoboCupサッカーシミュレーション2Dリーグの概要

2.1.　RoboCupサッカーシミュレーション2Dリーグ

ロボカップサッカーシミュレーション 2D リーグ（RoboCup2D）は，エージェントがコンピュータ内の仮想フィールドでサッカーゲームを行う競技である．RoboCup2D では，コンピュータ内に用意された 2 次元平面を仮想のサッカーフィールドとし，オリジナルの AI アルゴリズムを搭載したサッカーエージェント同士でゲームを行う．また，選手やボールの位置や速度はすべて 2 次元のベクトルで表現される．プレイヤーはキック，ダッシュ，ターンといった抽象化されたコマンドを基本動作とし，動作を選択することでサッカーを行う．ゲームは 3000 サイクルずつの 2 ハーフで構成され，合計 6000 サイクルとなる．1 サイクルは 0.1 秒単位で離散時間処理で進められる．

それぞれ独立してプログラムで選手とコーチは実装できる．選手ごとに視野が設定され，選手自身の視野内で認識された情報をもとに視覚情報としてサーバで管理され，それぞれの処理アルゴリズムで取得可能である．視覚情報には，他のプレーヤーやコーチから送られる情報も含まれており，フィールド上の離れている距離によって欠損などもある．得られた情報に基づいてシュート，ドリブル，パスなどの行動判断が各エージェントで行われる．しかし，視覚情報にはノイズや欠損等が含まれており，曖昧さを含む情報となる．どの程度正確であるかはエージェント側からは分からない．また，聴覚情報は確実に受信されるとは限らずサーバの状況やネットワークの状況によって欠損する場合もある．また，監督やコーチエージェントプログラムから送られるメッセージは通常のプレーでは時間的遅れがサーバ側に実装されているため，これらの揺らぎのある情報のダイナミックスを処理する AI 技術が各エージェントに求められる．ハーフタイムでは，その情報を即座に選手に伝えることができる．また，試合ごとに能力の異なる選手たちがランダムに与えられ，チームごとにポジションを割り振られる．ヘトロジーニアスエージェントを扱う AI 研究システムプラットフォームとして，多くの研究に利用されている．上記のように RoboCup2D には曖昧な要素があり，同じエージェントプログラムの対戦相手であっても，試合毎に試合内容やボールの保持率、支配率、得失点が異なると言われている [9].

2.2.　ロボカップサッカーシミュレータ

ロボカップサッカーシミュレータは、フィールド上のボールやプレイヤーエージェントの情報のシミュレーションを実行するためのシステムソフトウェアである．サーバクライアントシステムになっており，サッカーサーバーシステム (Soccer Server System) とも呼ばれている．サーバ上の仮想のフィールドで、ボールと選手の物理的な動きをシミュ

レートする．2D フィールであるため、ボールや選手が空中を舞うことはないように処理されている．図 2.1 にシミュレータの動作画面の一例を示す．様々な点で 2D に簡略化及び，抽象化された物理モデルが採用されており，実際の物理現象を忠実に再現して訳ではない．2D ロボカップリーグの特徴は，ロボットやセンサーの制御技術そのものではなく，マルチエージェントによるチームワークの AI アルゴリズム研究に重点を置いている [10].

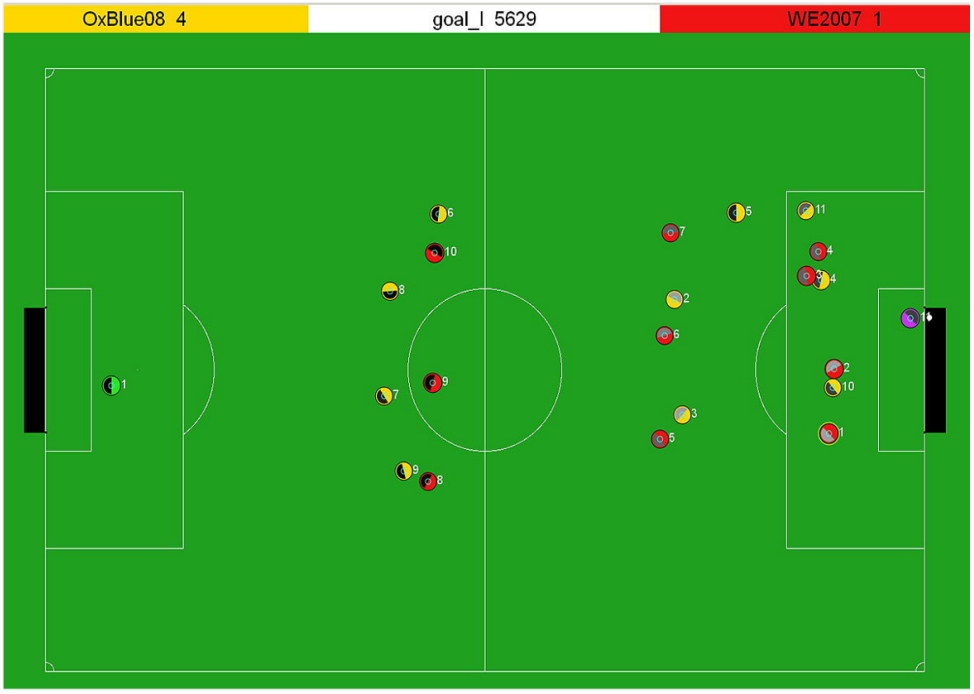


図 2.1: シミュレーターの実行画面

サッカーシミュレーターは、1 つのプログラムが動作しているのではなく、複数のプログラムが連動して動作している．複数のプログラムの処理が複雑に情報交換しながら進んでいくため，シミュレーション環境として提供される統合システムとなっている．このプログラムは，RCSoccerSim と呼ばれサッカーサーバとセットで用いて試合を進める RoboCup サッカーシミュションを行うために配布されている主なプログラムには次のようなものがあり，各開発者はこれらをダウンロードし，アレンジしながら開発を進める．これらのパッケージは RoboCup サッカーシミュションを動作させるために必要であり，各種プログラムのコンパイル方法や使い方，環境構築方法等の情報は [11] で取得できる．

- 2D サッカーサーバシミュレータ本体プログラム...rcssserver
- 画面表示専用プログラムシステム...rcssmonitor
- ログ再生用プログラムシステム...rcsslogplayer

シミュレーションを担当する rcsserver はサーバプログラムとなっている。ロボカップサッカーシミュレータは、サーバ・クライアント方式による分散マルチエージェントシミュレーションを実現している。サーバ・クライアント方式とは、プログラムをサーバとクライアントに分割し、各プロセスが役割を分担して処理を行う方式である。サーバーが情報を一元管理し、クライアントがその情報を利用するという、ネットワークプログラムではよくあるモデルである。サッカーシミュレータ RoboCup 上で動作するサッカーエー

ジェントは、rcsserver と通信を行うクライアントプログラムである。シミュレーション実行時の各プログラムの関係を図 2.2 に示す。

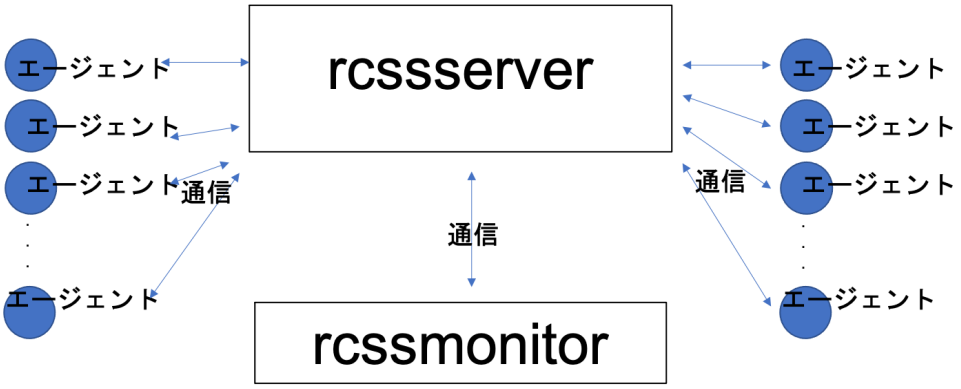


図 2.2: シミュレータ実行時のプログラムの関係図

サッカーエージェントは知覚情報となるメッセージを rcssserver から受け取り，それに応じて自分の行動コマンドのメッセージを rcssserver へ送信する．このメッセージの送受信を繰り返すことでフィールド上の物体の状態が変化し，シミュレーションが進行する．個々のエージェントが独立して通信を行う点も重要である．これは，サッカーエージェントは全て独立して制御されることを意味する．ひとつのクライアントプログラムが制御できるのはひとつのエージェントのみとなっている．そして，エージェント間の情報共有は完全な分散マルチエージェントシミュレーションが実現されている．

rcssserver 内部で使用されている座標系は左手座標系となっており，rcssmonitor でフィールドを表示する場合，フィールドの中心を原点とし，右を X 軸の正方向，下を Y 軸の正方向とする．角度計算もこれに基づいている．したがって，フィールドの中心から，右側のゴール方向を 0 度，真下の方向を 90 度，真上の方向を-90 度，左側のゴール方向を 180 度としている．角度を扱うときは，常に時計回りを正方向と考える．図 2.3 はフィールドの状態を示している．フィールドの大きさの設定は縦 105m× 横 64m である．フィールドの中心を原点とするため，各コーナーの座標はそれぞれ (-52.5,-34.0)(+52.5,-34.0)(-52.5,+34.0)(+52.5,+34.0) となる．各選手のプログラムは，この座標をもとにプログラムされる．

2.3. agent2d

本研究で用いるサッカーエージェントの AI プログラムは，agent2d をベースに開発をしている．agent2d は、RoboCup2010 で優勝したチーム HELIOS [12] の簡易版として位置付けられたサンプル AI エージェントプログラムであり，多くの研究者がこれをベースにしてマルチエージェントシステムの研究にもちいている．実装されているプログラムは，サッカーチームとして一定のパフォーマンスを発揮できるように設計・実装されており，ソースプログラムの整理されておりひな形として使いやすくなっている．agent2d を利用することで、rcssserver との通信や情報の統制や時間同期，各種センサー情報の解析やエージェント内部状態の更新，アクションコマンド生成などの下位層の処理を意識する

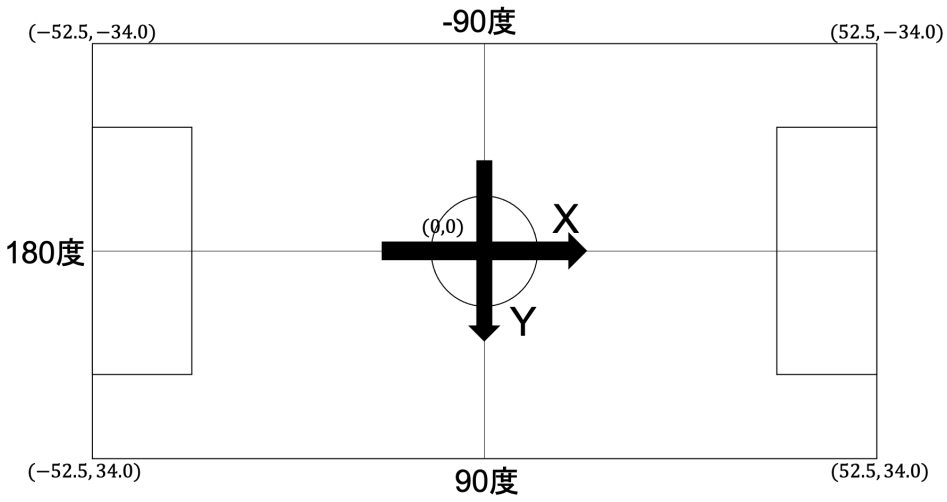


図 2.3: rcssserver の座標系

ことなく，エージェントの意思決定に向けた AI 実装を容易にしている [13]. 現在、サッカー シミュレーション 2D リーグの多くのチームが、agent2d のソースを手本にしてオリジナルサッカーエージェント開発を進めている．

2.4. アクション連鎖探索フレームワーク

アクション連鎖探索フレームワークは，各サッカーエージェントの行動決定を行うために行う探索木アルゴリズムである．このアルゴリズムにおける各時間単位の評価関数（状態価値関数）をファジィ推論を用いることで，$if\text{-}then$ 形式の知能化をし遺伝的アルゴリズムによってチューニングすることを本研究での主眼としている．そこで，agent2d に実装されている探索木の特徴を本節で説明する．

agent2d のプレイヤーエージェントは，トップダウン式の木探索によって協調的な行動計画を生成し，サッカーエージェントの意思決定を実現する。このモデルでは，ボールを蹴る時の行動計画を木構造で表現し，探索木を生成して探索することで，次のサイクルの行動を決定している．探索木のノードはサッカー場での行動選択後に予測される状態を表し，エッジではプレイヤーが選択可能な行動候補を表す．各ノードには状態に対する評価値が与えられる。プレイヤーは評価値が最大となるノードまでの行動連鎖を選択する．木探索の手法には最良優先探索が用いられている．

一般に用いられるゲーム木探索とのは異なり，トップダウン方式を採用している．トップダウン方式はボトムアップ方式に比べて，水平線効果によって読み間違いが発生しやすい欠点があるが，反面高速に処理できる利点がある．したがって，シミュレータの仕組みとして、1 サイクルの処理時間を 0.1 秒に設定しているロボカップサッカーにおいては，この方法が採用されている．そのため，各状態での評価値の計算品質が重要となる．また，ロボカップサッカーでは，時間サイクルを超えて行動をサーバに返した場合，スキップされて次のサイクルに処理が持ち越される。したがって、何もしないまま，そのサイクルが終了し，1 サイクル遅れの行動が実行される．これは、試合全体の実行時間を補償するためであり，各エージェントプログラムはこのサイクルを守らなければならない．つまり，

サーバとクライアントは非同期で動作し，行動命令をサーバが受け取るタイミングは個々でバラバラである．このため，より速い処理時間での木探索が要求される．

アクションプランは次のような手順で生成される．最初に，現在の状態が探索木のルートノードとする．次に，プレーヤーが観測した現在の状態と予測される状態をもとに，パス，ドリブル，シュートなどのアクションを生成する．このとき，対象の行動が実行可能かどうかを計算で判断し，相手にボールを奪取される可能性の高い行動に対して枝刈りを行う．そのため，実行可能と判断された行動のみが行動候補として扱われる．生成された行動と予測された状態は，評価関数によって評価される．行動，状態，評価値は探索木のそれぞれの子ノードとして格納される．評価値が最も高い子ノードを起点として，アクションの連鎖を形成する子ノードがさらに生成される．これを繰り返すことで，探索木が形成され，最も評価の高いアクションプランが実行される．木の深さが一定の値を超えると，子ノードから次状態のアクションを生成できなくなったり，シュートやクリアなどのアクションシーケンスの終わる事を意味するアクションが生成される．また，その子ノードを葉ノードとして扱うことができ，それらのノードを接続することで，探索木から行動列を生成することが可能となる．探索木の形成完了後，各評価値が最大となる行動列を行動計画として扱う．この行動計画により，サッカープレイヤーエージェントは戦略性の高い行動を選択することが可能となる．図 2.4 に探索木の一例を示す．簡略化のため，各アクションの評価値を丸印ノードで表し，アクションをエッジで記述している．この例で最終的に選択される行動は，評価値が最大となる (パス，パス，シュート) の行動列となる [9].

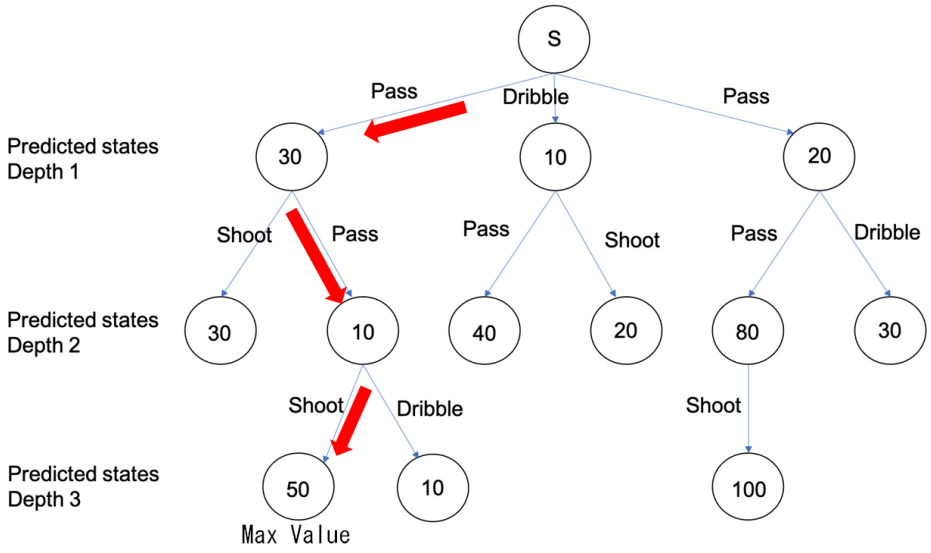


図 2.4: 探索木と局面評価関数の値の例

agent2d に実装されている標準的な評価関数の式を 2.1 式に示す．$U(x, y)$ を評価値，(x, y) を予測後のボールの位置座標，(x_{goal}, y_{goal}) を敵ゴール座標としている．この評価関数では，ボールの位置と敵ゴールとの距離を予測状態で計算して評価値としている．この評価の高い方向にむけてボールを移動させるようにアクション連鎖探索フレームワークは行動計画を立てて実行することになる．つまり，各エージェントはゴールに向かいパスやドリブルをしながらボールを進めることで敵陣に攻めて，得点を狙うようになっている．しかし，複数プレイヤによる意図的な協調動作を実現するには，自分の役割，敵味方の

配置状況，現在取るべき戦術などによって戦略的に評価値を変化させなければならない．標準的な評価関数を用いた時のフィールド上における評価値の分布をヒートマップをもちいて図 2.5 に示す．図のヒートマップでは，色が明るい色であるほど高い評価値を表している．したがって，フィールド上の全ての場所から一意に敵ゴールの中心に向けてボールを移動させる戦略をとる事になる．戦略としては，非常に単純であるため，本研究では *if-then* 形式で表現するファジィ規則 (ファジィルール) から推論値を算出することで効率的な AI による戦略を構築することを目指す．

$$U(x, y) = x + \max\{0, 40 - \sqrt{(x - x_{goal})^2 + (y - y_{goal})^2}\} \tag{2.1}$$

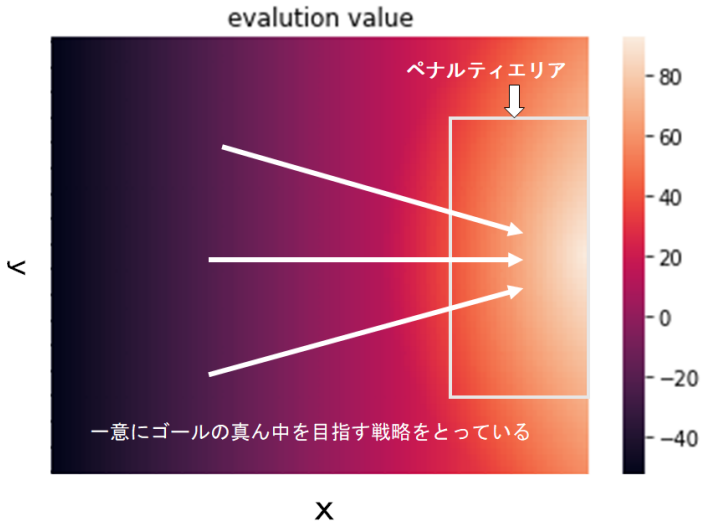


図 2.5: フィールドにおける評価値の分布

第3章　JapanOpen2020大会の試合分析と評価指標の選定

3.1.　大会の各チームの分析

RoboCup2D では定期的に大会が開催されている．2020 年 8 月 19 日に開催された夏の大会と 2020 年 10 月 29 日に開催された JapanOpen2020 の全試合 (100 試合) のログファイルを分析し今後の実験の指針にすることを検討した．

3.2.　試合の分析

今回収集したログファイル 100 試合分から，LogAnalyzer3（付録 A.3 参照) を使って解析した．強いチームの定義をより多く点が取れるチームとした．これより，1 試合の点数との相関関係を分析した．相関係数に正の相関が見られた場合，点を多くとるにつれてその値も大きくなるということが認められるため，強いチームの条件と言える．100 試合分のログファイルから 1 試合の得点と正の相関があるものを散布図と相関係数から判断した．相関係数の計算方法としては，$k_1 \cdots k_n$ を自チームの得点，$l_1 \cdots l_n$ を LogAnalyzer3 によって算出される指標とする（全ての指標は付録 B 参照)．LogAnalyzer3 によって算出される指標を l として一つの指標ごとに得点数との相関を見る．$\overline{k}, \overline{l}$ はそれぞれの平均，s_k は k の標準偏差であり，式 3.1 に示す．s_l は l の標準偏差であり，式 3.2 に示す．s_{kl} は共分散であり式 3.3 に示す．そして r_{kl} を相関係数として式 3.4 によって与えられる [14].

$$s_k = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(k_i - \overline{k})^2} \tag{3.1}$$

$$s_l = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(l_i - \overline{l})^2} \tag{3.2}$$

$$s_{kl} = \frac{1}{n}\sum_{i=1}^{n}(k_i - \overline{k})(l_i - \overline{l}) \tag{3.3}$$

$$r_{kl} = \frac{s_{kl}}{s_k s_l} \tag{3.4}$$

3.3. 分析結果

評価指標と自チームの得点との相関係数を表 3.1 に示す．正の相関が見られた評価指標は主に自チームのスルーパス数, 自チームのシュート数, 自チームの支配率, 敵チームのペナルティエリアへの侵入数であった．負の相関係数が見られる評価指標もあるが，全て敵チームに依存しているため，今回扱わない事とした．それぞれの散布図は図 3.1 である．

表 3.1: 評価指標と自チームの得点との相関係数

評価指標	相関係数
自チーム: スルーパス数	0.26
自チーム: シュート数	0.97
自チーム: 支配率	0.38
敵チーム: ペナルティエリアへの侵入数	0.88

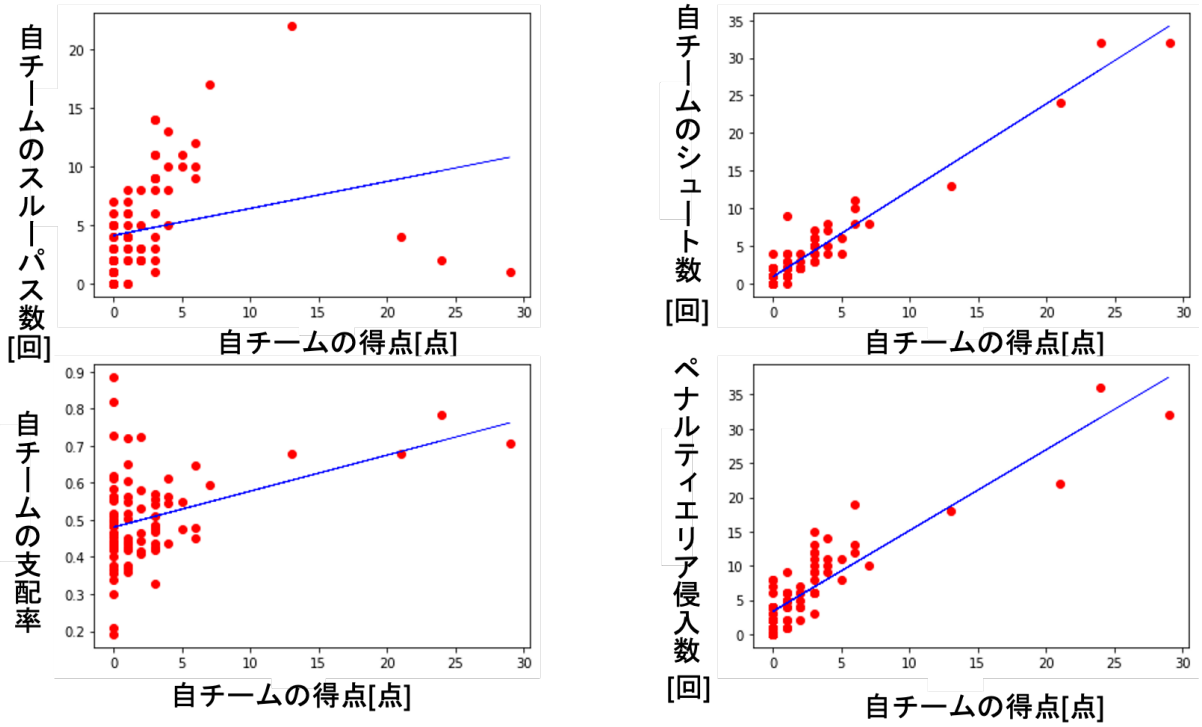


図 3.1: 変数それぞれの散布図

この結果から，非常に強い正の相関が見られたシュート数とペナルティエリアの侵入数は多いければ多いほど得点を取ることに繋がっているということが見て取れる．やはり，より多くの得点を取るチームはペナルティエリアに侵入しシュートチャンスを多く作っているチームということを分析結果からも分かった．よって，シュート数とペナルティエリアへの侵入数は向上させるべき評価指標と考えて実験を進めていくこととする．

また，スルーパスの数では強い相関とは言えないが，図 3.1 から見て取れるように，スルーパスの数は 1 試合の点数が 15 点以内のみで見ると，より多く点数を取るチームであるほどスルーパスを多く出していると言える．つまり強さの差が圧倒的に開いているチー

ムではスルーパスは必要としないが，強さの差に開きがあまりないチームでは必要な強さに関係する指標だと言える．また支配率に関しても，図 3.1 から，点を取れていないチームが，フィールドの中盤の位置でボールを支配して支配率が高くなっていることもあるが，点を多く取るチームで支配率が低いチームがないので，強さに関係する指標だと言える．しかし，本論文における実験内容では行動パターンを選択する際に直接的なパスを選択するか，スルーパスを選択するかは状況によって変化するので実質的にスルーパス数を向上させることに繋がらないためスルーパス数は考慮しない．支配率に関しては向上させることができるのだが，支配率を向上させようとした場合ボールを中盤で保持しボールを取られる可能性が高くなる前線にボールを運ぼうとしない動きが見られた．これは得点には繋がらない動きである．チームの強さにおいて最も重要である得点力の向上に繋がらないと判断し，本論文では得点数，シュート数，ペナルティエリアの侵入数を上昇させるべく実験を行っていくこととする．

3.4. agent2d の性能

評価関数を次章から設計していくが，設計した評価関数がどれほどチームの強さに影響を与え，どのような変化が出たのかを検証するためにも，比較対象として既存の agent2d がどれほどの強さを持っているかをまずは知る必要がある．今回は既存の agent2d を 100 試合戦わせ，LogAnalyzer3 を使って，得点数，シュート数，ペナルティエリアの侵入数の平均を求め，比較対象とする．本論文では評価関数の設計の際スルーパス数，支配率を比較対象としないため，スルーパス数と支配率は求めない．また，前半のみの 3000 サイクルとしている．

表 3.2 より既存の agent2d の成績を見ると，1 試合（前半のみ）の得点は約 1 点であり，シュート数とペナルティエリアの侵入数を見ると標準偏差が 1 を超えているため平均から多少ずれた値がでる場合もあるが，平均の値と考える．

表 3.2: 変更なしの agent2d の成績

評価指標	平均 ± SD
チームの得点 [点]	1.12 ± 0.99
チームのシュート数 [回]	2.68 ± 1.42
ペナ侵入数 [回]	5.61 ± 0.51

第4章　フィールド座標を用いたサイド攻撃に特化した評価関数の実装

4.1.　既存の評価関数の問題点

3章より，向上させるべき評価指標を得点数，シュート数，ペナルティエリアの侵入数とした．そのためそれらの変数を向上させるために評価関数を考えていくことになる．しかし，agent2dの試合を目視解析したところサイド攻撃が単調であることにきづいた．

図4.1の中でボールを保持しているエージェントは，10番のエージェントである．10番はサイドからボールを保持したまま敵ゴールに近づいていく，ゴールに最も近づいた状態では11番のエージェントがフリーになっており，11番にパスをできれば攻撃の展開がかなり広がると考えられる．しかし，既存の評価関数ではボールと敵ゴールの距離のみを見ているので，ボールとゴールの距離が大きくなるような行動を行うことはない．よってそのままシュートするかゴールに近づきボールを奪われるという結果となる．よって，このような状況に陥った場合にゴール前のフリーのエージェントにパスを回すことができればサイド攻撃がさらに活性化し，強さにも関係するのではないか，そして向上させるべき評価指標も向上するのではないかと考え，評価関数を設計した．本章のタイトルである縦方向の位置とは，サッカーフィールドのy座標であり，このy座標を用いることで，既存の評価関数に変更を加える．

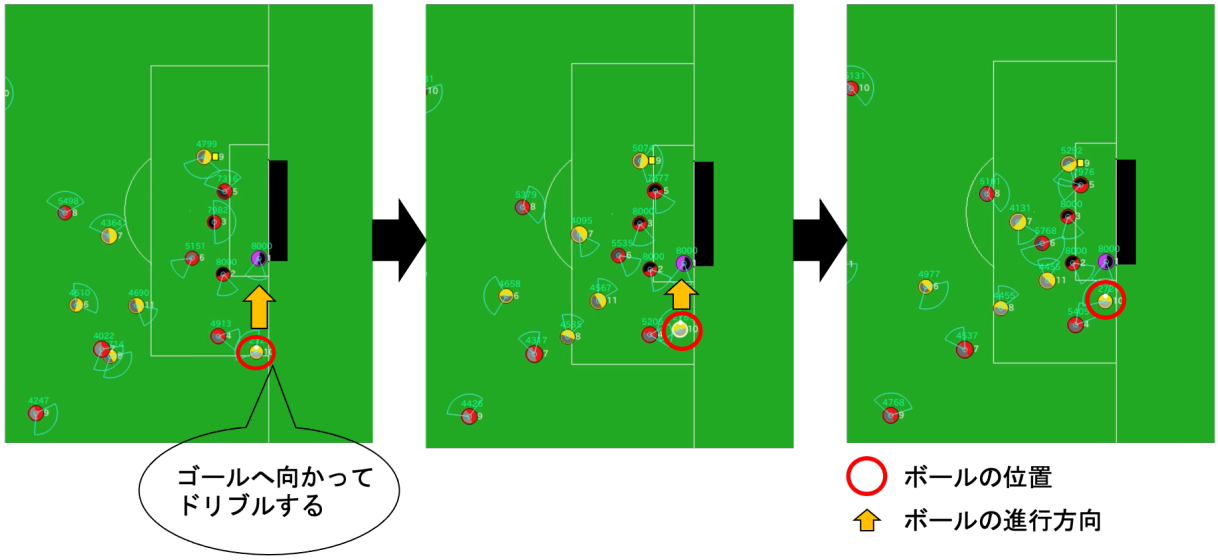


図 4.1: 既存の agent2d のサイド攻撃

4.2. y 方向座標情報を用いた評価関数の設計と数値検証

サイド攻撃の場合のみに限って変更するため x_{now} を現在のボールの位置とすると $x_{now} \geq 36.0$ の場合（図 4.2 参照）のみ変更した評価関数を適用する．$x_{now} < 36.0$ の場合は既存の評価関数である式 2.1 に新たに y 方向のパラメータで制御する r を導入することでペナルティエリアの左右側にボールを集めて，アクション連鎖探索がサイド攻撃を誘発するように設定した．

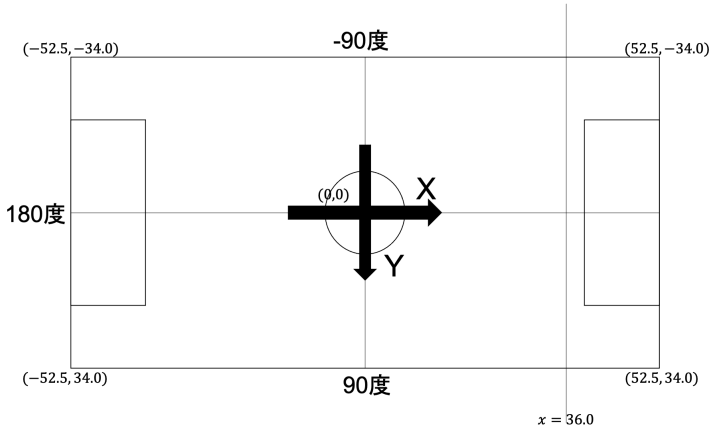


図 4.2: 評価関数の変更範囲

$$r(y) = \frac{0.3|y|}{34.0} \tag{4.1}$$

$$U(x, y) = x + \max\{0, 40 - r(y)\sqrt{(x - x_{goal})^2 + (y - y_{goal})^2}\} \tag{4.2}$$

$U(x, y)$ を評価値，(x, y) を予測後のボールの位置座標，(x_{goal}, y_{goal}) を敵ゴール座標とする．r を重みとして式 4.1 より求める，分母の値である 34.0 は y 座標の最大値である．つまり r の値は予測後のボールの y 座標の絶対値が小さい値であるほど r の値も小さいものとなる．よって式 4.2 より予測後のボールの位置座標と，敵ゴールの位置座標のユークリッド距離を求めている部分に重み r をかけることにより，ボールとゴールの位置が多少離れたとしても，y 座標の絶対値が小さければ高い評価値になるという評価関数を設計した．

図 4.3 は設計した評価関数におけるフィールドでの評価値の分布である．色が明るい色であるほど評価値は高くなっている．図 2.5 と比較すると図 2.5 では敵ゴール付近が最も評価値が高くなっており，全体を見ると x 座標が高い値であればあるほど評価値が高くなっている．図 4.3 では，敵ゴール付近が最も評価値が高くなるのは同じである．しかし，敵ゴール付近の評価値と敵ゴールから少し離れているかつ，y 座標が 0 に近い部分の評価値が同等の値となっている．よってサイドから味方がドリブルで攻め入り，敵ゴールに近づいた状態でも，ゴールから少し離れた位置に，フリーのエージェントが存在すれば，パスを出すという可能性がある．

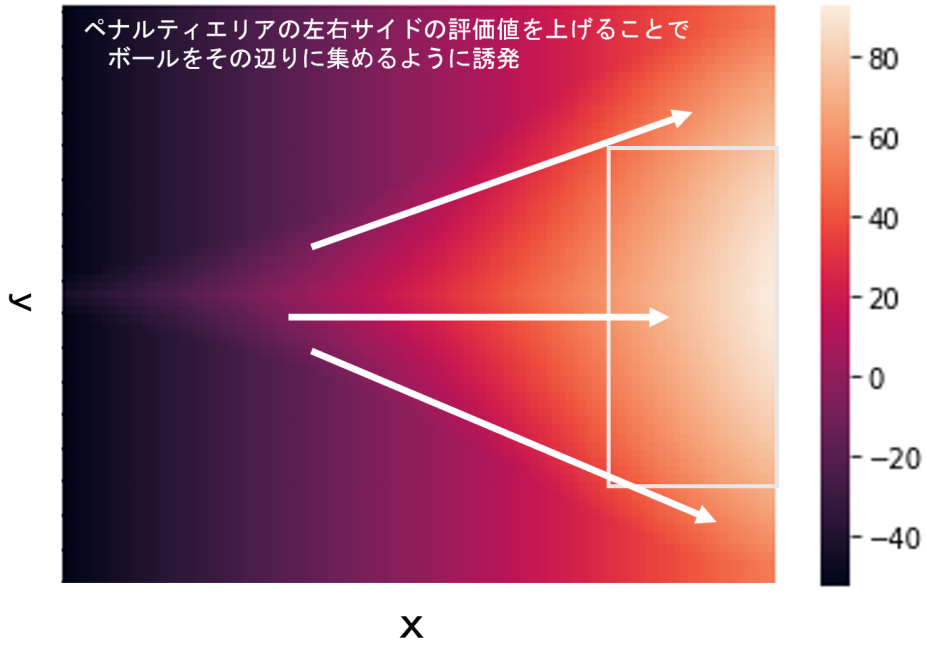


図 4.3: フィールドにおける評価値の分布

4.3. 結果

図 4.4 は，評価関数変更後の agent2d のサイド攻撃の一部である．既存の評価関数では，サイド攻撃ではパスをすることなくドリブルのみで相手ゴールに近づく行動のみであった．しかし，図を見て分かるように，今回設計した評価関数では，10 番のエージェントがサイドからペナルティエリアに侵入すると，11 番のエージェントが，フリーでありパスを出せると判断した場合，10 番のエージェントは，11 番にパスをだし，11 番のエージェントは，ゴールを向くとともに，シュートを決め，得点できていた．この一連の動作は評価関数を変更したことにより選択できていた．

この結果から，これまでの agent2d のサイド攻撃が単調であるという弱点を克服し，サイド攻撃の組み立ての選択を広がったということを実験観察により確認した．図 4.5 は，図 4.4 で選択された，アクション探索木で生成された行動を表している．

表 4.1 は，設計した評価関数を既存の agent2d のプログラムに適用したものと既存の agent2d を 100 試合（前半のみの 3000 サイクル）戦わせた結果のそれぞれの平均と標準偏差である．また，比較対象として表 3.2 の結果も表示している．図 4.4 のような行動が確認でき，チームの強さに関しても，既存のチームより強くなったのではないかと予想したが，表 4.1 から見て取れるように，得点能力の部分ではほぼ差が現れることがなかった．またシュート数の部分でも多少の誤差はあるものの差は現れなかった．変わったと言えるのは，ペナルティエリアへの侵入数である．標準偏差の差が多少の誤差であるが平均では差がでているためペナルティエリアへの侵入力に関しては，能力が向上した．

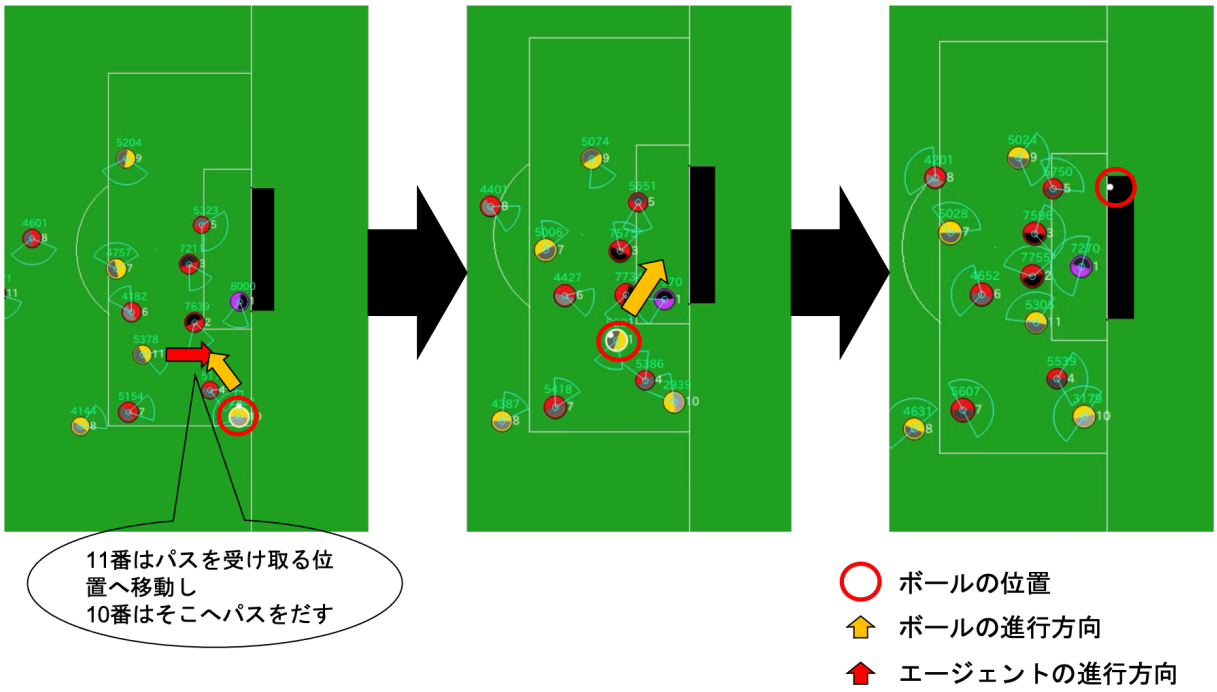


図 4.4: 変更後のサイド攻撃

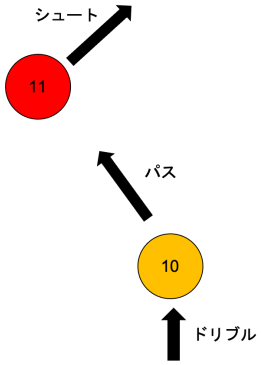


図 4.5: 探索木で選択された行動

表 4.1: 各チームによる 100 試合の対戦成績

	対戦評価					
	得点数 [点]		シュート数 [回]		ペナルティエリア侵入数 [回]	
各チーム	平均	±SD	平均	±SD	平均	±SD
本章	0.99	±1.18	2.36	±1.03	7.19	±1.43
agent2d	1.12	±0.99	2.68	±1.42	5.61	±0.51

4.4. 考察

本章では，評価関数の設計をボールの縦方向の位置を用いて行った．目的としてはサイド攻撃の単調な部分にさらに選択肢を与え，攻撃のアイデアを増やすというところであった．その目的に関しては，達成した．またそのように設計した評価関数によりチームの強さの向上にも影響を及ぼすことができるであろうと考えていた．しかし表 4.1 からも見て

取れるように，一部分での向上は見られたが，チームの強さが既存のものより上がったとは言えないことが分かる．つまり，サイド攻撃のパスやドリブルのアイデアは増え，ペナルティエリアへの侵入力は上がったが選択した行動が得点数の上昇には至らなかった．このように人間が値を調整し評価関数を設計することは，一部分の問題解決には繋がるが，実質的にチームの強さに影響を及ぼすことは難しいと考えた．

第5章　遺伝的アルゴリズムを用いたファジィ推論評価関数の最適化

5.1.　遺伝的アルゴリズム

遺伝的アルゴリズム（GA）は，1975年にミシガン大学のJohn Henry Hollandが「Adaptation in Natural and Artificial Systems」論文で紹介したアルゴリズムで，生物の進化プロセスを計算に導入することで航空機用タービンや集積回路などの複雑なデバイスの設計における機械学習技術の利用まで、複雑な適応システムの設計や最適化に役立つことを目指している．その研究の多くは1960年代から試されており、パラメータ群の最適組合せや最適な値をシミュレーション的に探索・発見できる代表的なアルゴリズムである [15]．GAは，生物の進化の原理にヒントを得た近似解探索のアルゴリズムであり，特に生物進化で行われている、環境に適応した種の生き残り、環境に適応できない種の死滅の過程を数値シミュレーション敵に繰り返すことで最適な解探索を実現している．例えば，寒冷地では生物は生き残るために寒さに適応する．しかし，中には寒さに耐えられず絶滅してしまう生物もいる．図5.1はその一例であり，GAでは対象の環境に対する適応度から，優秀な個体 (パラメータ集合) を次世代に伝え，そうでない個体は排除するという考え方に基づいている．繰り返した後の最終世代になると，最も適合度の高い個体の遺伝情報が選択される。これは，最も適合度の高い個体集合が生き残ったことになり，最適な解となる．問題点は，繰り返し世代毎に環境に対する適合度を計算しなければならないことであり，本研究においてはサッカーの試合を1試合行うことと同じになる．実際の実験では，サッカーサーバの設定を使って事実上の試合時間サイクルを小さくして実験しているが，それであっても一世代の評価に時間がかかるため，クリアする難しい技術的課題である．

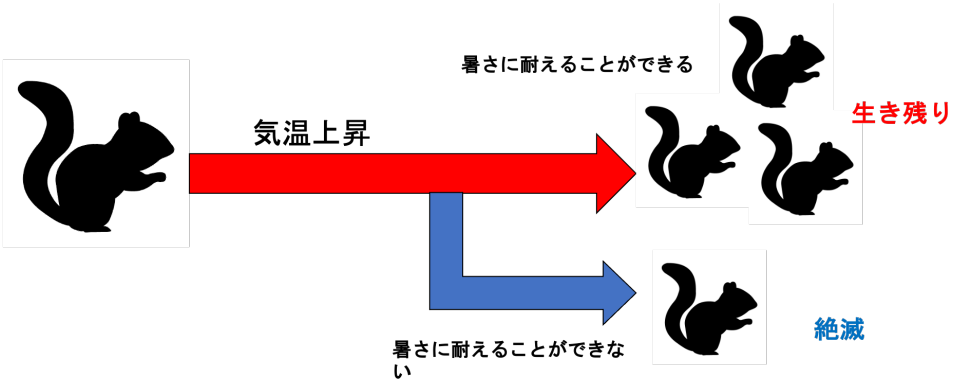


図 5.1: 進化の例

照明デザインや，音声／画像処理・検索，新幹線N700系の先頭形状デザイン [16] などにもGAは利用されている．GAアルゴリズムでは、数字列や文字列で表された情報の仮

想な遺伝子列を集合とした染色体（Chromosome）を構成する．この情報集合を遺伝子として扱って，この各遺伝子の組み合わせに対する遺伝子型の個体を対象に探索したい解空間に写像することできるようにしておく．これを表現型と呼ぶ。これらの染色体に対して、個体の優劣を決める評価基準となる適合度関数（Fitness Function）が定義される。そして、個体の適合度に応じた選択を行う。さらに、交叉、突然変異などの遺伝子操作を行い、新しい個体を生成する。このような遺伝子操作と適合度の評価を繰り返すことで、より適合度の高い個体が生成される [17]。図 5.2 に Genetic Algorithm のフローを示します。

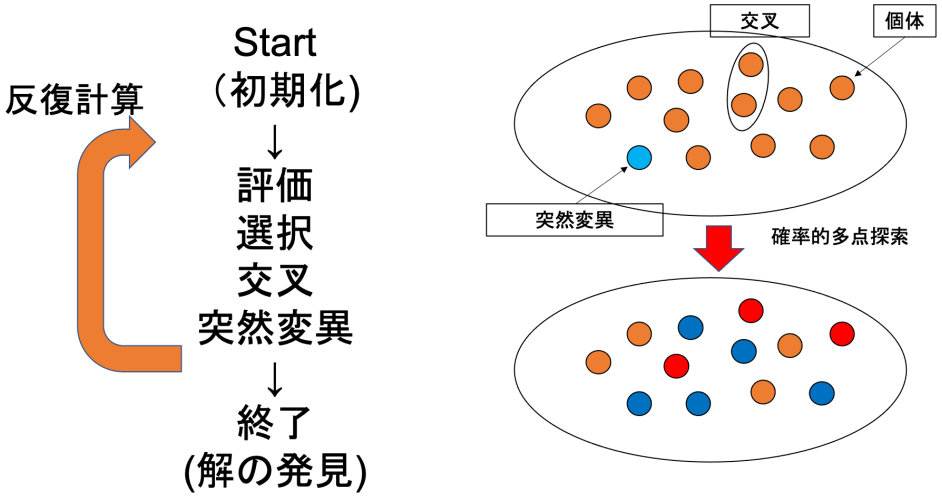


図 5.2: 遺伝的アルゴリズムのイメージ

5.2. 実数値遺伝的アルゴリズム

実数値遺伝的アルゴリズム（Real-coded Genetic Al-gorithms：実数値 GA）とは，遺伝子の表現を実数値型とした遺伝的アルゴリズムである．5.1 節で述べた GA は遺伝子の表現を 0，1 のバイナリ型としたものであった．現実世界の問題では，バイナリ型の組み合わせのみではなく，実数値で表現される組み合わせも多く存在するため，このような場合に遺伝子を実数値として取り扱う実数値 GA が必要となる．

基本的な動作の流れは GA と同じであるが交叉，突然変異の手法は遺伝子の表現が実数値であることを考慮して，変更する必要がある．しかし実数値 GA の場合，交叉が突然変異の役割も担っており，突然変異は補助的な役割として扱われる．そのため突然変異が解探索において悪影響を及ぼす場合もあるので，突然変異を使用しないという選択肢もある [18].

交叉

遺伝子を構成する個体は，長さを m とすると m 次元のベクトルとなる．親個体が状態空間上に離れて存在している場合は，子個体を広い範囲で生成しなければならないが，進化が進み各親個体が優秀になり，状態空間上の近い範囲で存在している場合は，子個体は親個体の近傍に生成することが望ましいと考えられる．このようなことを考慮して，交叉法は設計されている．

突然変異

実数値 GA の場合は，以下のような変異方法が使用される．

- 一様突然変異（Uniform mutation）

  設計変数ごとに，実行可能領域内に一様乱数により，新規の実数値を発生させる．

- 境界突然変異（Boundary mutation）

  一様突然変異とほぼ同じであるが，新規の値は実行可能領域の境界上にとる．

- ガウス突然変異

  個体の遺伝子に，設定した平均値と分散のガウス分布から，発生させた乱数を加算する．

前述したように実数値 GA では，交叉が突然変異の役割もしているため，突然変異を使用しないということも考えられる [19].

5.3. ファジィ推論

ファジィ理論は，ファジィ集合とファジィ推論からなり，身の回りの曖昧な情報を扱うための理論である．その特徴として，設計者の知識をコンピュータに応用する際に，人間の感覚に近い自然な表現で知識を記述できる手法として，産業界で広く定着している．ファジィ集合から if-$then$ 形式のファジィ規則を用いたファジィ推論手法が考案されている．このファジィ理論を制御に応用したファジィ制御は，様々な分野で研究開発が行われている [20]．また，心理学や経済学などの文系分野から，プラントや自動車，家電製品などの制御といった工学系分野まで幅広いく応用されている．

ファジィ制御で用いられるファジィ推論は、ファジィ論理とファジィ規則 (ファジィルール) に基づいて結論や制御変数を推論し，決定する手法である．入力情報をファジィ集合と照合し，必要な制御操作変数を決定することで実現する．主にプロセス制御、エキスパートシステムなどに応用され、家電製品にも当たり前のように使われている．ファジィ推論では，通常、if(前件部)$then$(後件部) 型のファジィルールが用いられる。一般的な制御方法とは異なり，ファジィルールの前件部と後件部は，それぞれファジィ集合で記述できる．大きい」「弱い」など定量的・決定的に定義できない内容を表現することが可能で，”車のスピードが速かったらスピードを落とせ！ ”など、人間の経験的知識を規則 (ルール) として表現することが可能になる．1980–2000 年までの AI(人工知能) では，このように人間の知能や規則を規則 (ルール) 化することで様々なアプリケーションに応用してきたため，ファジィ制御のような技術がこの時代に盛んに研究され，製品などに実装がなされてきた．

本研究で用いているファジィ推論には，簡略型推論法を用いている．簡略型推論法は，菅野氏 [21] らによって提案されたもので，基本的には Min-Max 重心法と同じように重みバランスによって推論結果を得る手法である．簡略側推論法では，推論計算時に，前件部ではメンバーシップ値の Min 演算を行い，後件部ではファジィ集合ではなく，定数（シングルトン）を用いる．メンバーシップを後件部に用いた場合は重心を求める過程で面積

を計算しなければならないが，簡略型では固定値の高さ 1.0 のシングルトンとの Min 演算 (掛け算) と足し算だけで求めることが可能であり，計算処理上高速である．また，単純な条件下であれば，簡略型推論は重心法や台数加算重心法などの後件部にメンバーシップ関数を用いた手法と推論性能が変わらないと言う報告があり，計算速度を重視する本研究では，この簡略型推論法を用いる．また，前件部のメンバーシップ関数の設計はサッカーフィールド固有の値と人間の知能を含めたルールを設定することとする．また，先の GA では，規則の前件部に対応した後件部シングルトンの定数 (位置場所) を最適化するように探索することで評価関数の探索向上を目指す．

評価関数の処理部分では，agent2d の評価関数の計算に，より処理速度速いものが求められるため，計算速度の早い簡略型推論法を用いた．また，簡略型推論法の後件部のシングルトンにおけるパラメータは，実数値であるため実数値 GA を用いてチューニングした．

本研究では，アクション探索フレームワークの評価関数の計算で，簡略型推論法を用いる．また今回の評価関数の設計においても，4 章と同様に x_{now} をボールの現在の x 座標とすると，設計する評価関数を $x_{now} \geq 36.0$ の場合のみ適用することとする．それ以外の場合は式 2.1 の既存の評価関数を使用する．図 5.3，図 5.4 は今回設計した評価関数の簡略型推論法における前件部と後件部である．

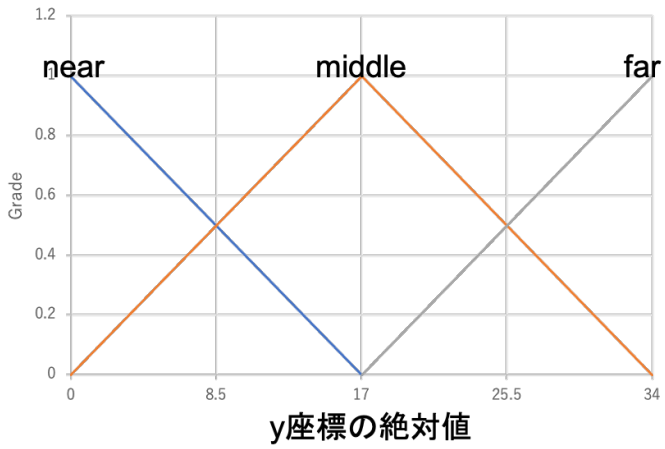


図 5.3: 設計した評価関数の簡略型推論法における前件部

簡略型推論法の前件部における図 5.3 のメンバーシップ関数では near，middle，far の 3 つのルールを定義した．y はアクション連鎖探索で予測した状態における y 座標の絶対値である．また図 5.4 は後件部におけるシングルトンである．シングルトンのラベル値を r_1^* , r_2^*, r_3^* とする．式 5.1 のように 3 つのルールを作成した．これは毎回予測後の y 座標の絶対値を取得して，最終的に r の値を求めるものである．r の計算方法としては式 5.2 である．そして式 5.3 に示すように，r の値を用いてアクション探索木の評価値を算出する．$U(x, y)$ は評価値である．

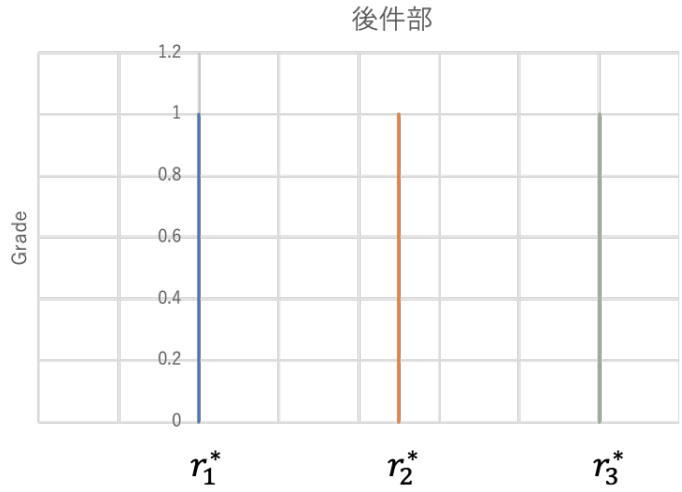


図 5.4: 設計した評価関数の簡略型推論法における後件部

$$
\begin{aligned}
R_1 &: If \quad y \quad is \quad \text{near} \qquad Then \quad r \quad is \quad r_1^* \\
R_2 &: If \quad y \quad is \quad \text{middle} \qquad Then \quad r \quad is \quad r_2^* \\
R_3 &: If \quad y \quad is \quad \text{far} \qquad Then \quad r \quad is \quad r_3^*
\end{aligned}
\tag{5.1}
$$

$$
r = \mu_{"near"}(y) \times r_1^* + \mu_{"middle"}(y) \times r_2^* + \mu_{"far"}(y) \times r_3^* \tag{5.2}
$$

$$
U(x,\, y) = x + \max\{0,\, 40 - r\} \tag{5.3}
$$

5.3.1. パラメータのチューニング

上記で記述した簡略型推論法の後件部におけるシングルトンのラベル値 r_1^*, r_2^*, r_3^* を実数値 GA を用いてチューニングし，チームの強さを向上させる． 今回の実数値 GA の手順としては，図 5.5 の通りである．また今回の実数値 GA の交叉と突然変異の方法としては，交叉は BLX-α 交叉を使用し，突然変異は，ガウス変異を使用した．交叉と突然変異の手順としては，図 5.6 に示すように，まず親個体から最も適応度の高い個体を残し，それ以外の個体で交叉確率に基づき交叉を行い，最大適応度の個体以外で，設定した突然変異確率で突然変異を行うという流れである．

今回の実数値 GA で設定した各パラメータと選択，交叉，突然変異の方法は以下の通りである．

- 繰り返し世代数：100
- 集団内の個体数：15

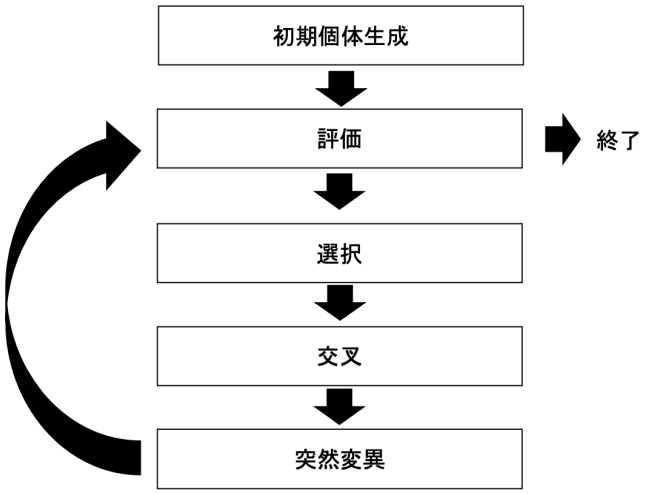


図 5.5: 実数値 GA の手順

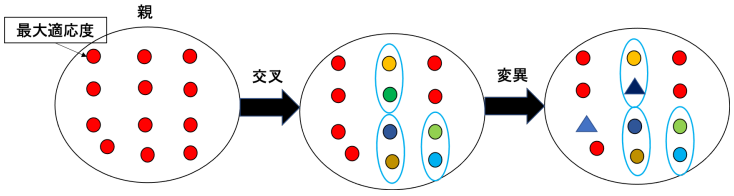


図 5.6: 実数値 GA の交叉と突然変異の手順

- 個体の次元数：3
- 選択方法：トーナメント選択
  - トーナメントサイズ：4
- 交叉方法：BLX-α 交差
  - α ：0.2
  - 交叉確率：0.5
- 突然変異方法：ガウス変異
  - ガウス分布：平均 0，標準偏差 10
  - 突然変異確率：0.2
- 試合サイクル：3000 サイクル（前半のみ）

各個体の評価方法を式 5.4 に示す．$fitness$ は個体の適応度，$shoot$ は 1 試合での自チームのシュート数，$penalty$ は 1 試合での相手チームのペナルティエリアへの侵入数である．これは個体ごとに毎回 3 試合行い，サッカーサーバーが出力したログファイルを LogAnalyzer3 を用いて，ゲームログを解析し 3 試合分のシュート数とペナルティエリアの侵入数から個体の適応度を計算した．$penalty$ に 0.2 の重みを与えているのは，1 試合のシュート数とペナルティエリアの侵入数では，ペナルティエリアへの侵入数のほうが数値がかなり大きくなるため，重みを与えることにより 2 つの数値を向上させていける適応度とした．

$$fitness = \sum_{i=1}^{3}(shoot_i + 0.2penalty_i) \tag{5.4}$$

5.3.2. 結果

0 世代から 100 世代の世代と各集団の最も適応度が高かった個体の適応度を図 5.7 に示す．図から世代が進んでいくにつれ最大適応度も上昇していっていることが分かる．これにより，各集団で最大の適応度をもつ個体を残していることが確認できた．また，実数値 GA によって適応度を上昇させれていることがわかった．

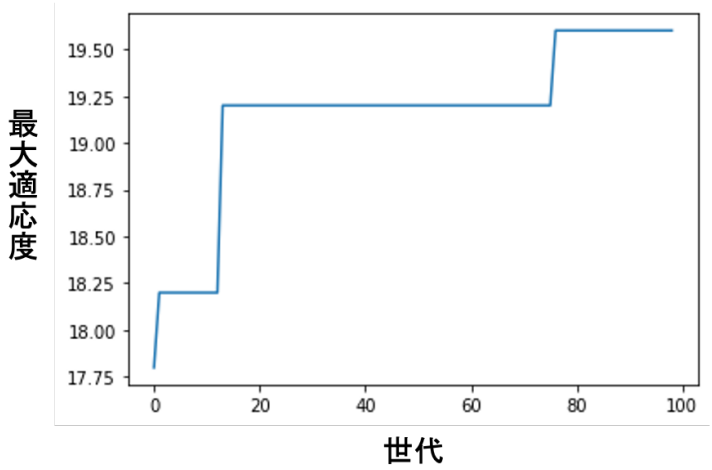


図 5.7: 各世代の最大適応度

実数値 GA によってチューニングした値を使用し設計した評価関数を取り入れているチームと，既存の agent2d で 100 試合試合を行った（前半のみの 3000 サイクル）．100 試合分の評価指標の平均と標準偏差を表 5.1 に示す．また，比較対象として表 3.2 の結果も表示している．図 5.7 からも見て取れるように適合度が上昇しているため，シュート数，ペナルティエリアの侵入数が向上し，得点能力も向上するという期待があったが，既存の agent2d と比較して向上していると言えるのは，ペナルティエリアの侵入数のみであった．得点率とシュート数は既存の agent2d よりも下回っていた．また，試合でのエージェントの動きを図 5.8 に示す．図に示すように，10 番のエージェントはペナルティエリアにボールを保持しながら出入りを繰り返す動きをしていた．ペナルティエリアの侵入数はペナルティエリアのラインをボールが通過することによりカウントされるので，この動きにより試合でのペナルティエリアへの侵入数も多いことがわかった．

5.3.3. 考察

本章では，評価関数の設計をファジィ推論方の一種である簡略型推論法を用いた．また，その後件部のラベル値を実数値 GA を用いてチューニングした．これにより，人間の調整ではなく，人工知能を用いることによるサイド攻撃のさらなる強さの向上とそれに

表 5.1: 各チームによる 100 試合の対戦成績

	対戦評価					
	チームの得点数 [点]		チームのシュート数 [回]		ペナ侵入数 [回]	
各チーム	平均	±SD	平均	±SD	平均	±SD
本章	0.44	±0.61	1.14	±1.03	13.39	±4.70
agent2d	1.12	±0.99	2.68	±1.42	5.61	±0.51

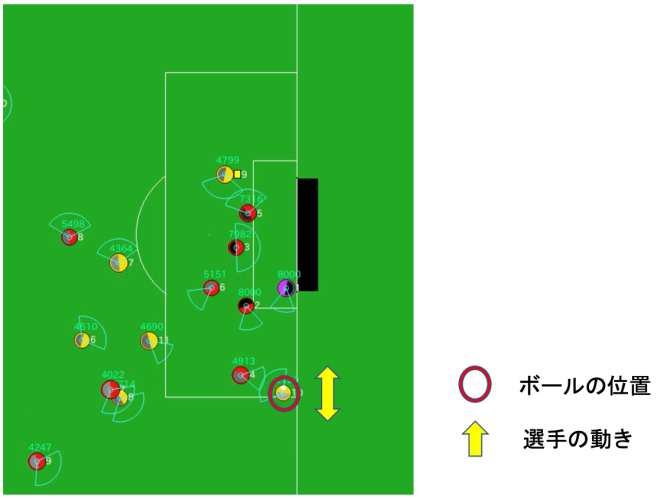


図 5.8: エージェントの動き

伴ったチーム全体の強さの向上が可能だと予測したが，実際には表 5.1 に示すようにチーム強さは向上しなかった．

今回の実験で不十分であった点としては，ペナルティエリアの侵入数のみを向上させる適応関数となっていることが挙げられる．本章ではペナルティエリアの侵入数とシュート数を向上させることで，シュートを多く打ち得点を多くとることができるという期待のもと実験を行ったがそのようには，至らなかった．重み調整などを再度調整し，ペナルティエリアへの侵入数のみに偏らない適応関数を設定していく必要があると考えた．また，本章でも前章でも設計した評価関数を適応したのは，サイド攻撃の場合のみである．サイド攻撃のみに設計した評価関数を適応しても，期待できるほどのチーム能力の向上はみられないという可能性もある．本章での GA の設定した繰り返し世代数や個体数は実験の時間が長くなりすぎるため，本来 GA で扱うパラメータ数よりも少なくしている．多大な時間を使って，より多くサイクルを回していけば多少の向上が得られる可能性も否定はできないが，サイド攻撃のみに設計した評価関数を適応するのではなくフィールド全体に評価関数を適応することによって，GA の探索範囲も広がりより得点に繋がる評価関数を効率よくチューニングできるのではないかと考えた．次章からは，本章での実験結果を活かし，さらにチームの得点力を上昇できるように実験を行っていく．

第6章　評価対象をフィールド全体に変更した評価関数の設計と実験

本章では，5 章の評価関数からさらに進展させるため，フィールド全体に適応できる評価関数の設計を行った．5 章と同様に遺伝的アルゴリズムとファジィ推論法を使って，評価関数を最適化した．

6.1.　フィールド全体に適用した評価関数の設計

6.1.1.　効率的に実験を行うための試合条件

4 章，5 章では，GA における個体の評価や，チームの強さを数値で得るために毎回試合を行っていた．その試合条件としては，3000 サイクル（前半のみ）の試合を行い，各チームの強さを評価していた．チームの強さを図る上で試合を行うことは重要であり，必ず行わなければならないものなのであるがこれには問題点があった．一つは，攻撃を行う回数が毎試合ごとに変化があるということである．本論文では，チームの強さを上昇させることを目的とおいているが，チームの強さとは試合においていかに多く得点できるかというところをチームの強さと考えている．そのため守備の部分は本論文では考慮していない．よって攻撃時においていかに得点できるかというところを試合において判断したい部分ではあるのだが，1 試合において攻撃回数にばらつきがあると等しい評価ができていないのではないかと言う問題点が挙げられる．二つ目は，単純に試合に時間がかかりすぎるという点である．チームの強さを向上させるために GA を使って 100 世代数における実験をおこなっており，その過程において個体の評価のために何万回にもおける試合を行う必要があるのだが，試合の時間がかかりすぎると実験時間も大幅に膨れ上がる．そのため効率よく試合をして評価を行う必要があった．このような問題点の改善のために，試合条件を設定することとした．試合の流れとしては，図 6.1 の通りである．初めにボールの初期位置をランダムに設定し配置する，その位置から味方フリーキックからのスタートとし，攻撃を始める，200 サイクル経過すれば一度ゲームをストップし，再度ランダムに配置されたボール位置からフリーキックによってゲームを再スタートするというような条件である．この試合条件によって，毎試合ごとに等しい回数攻撃を行うことができ，より正確にチームの得点力を数値化することが可能になると考えれる．また，図 6.2 に試合条件の説明を表示している，図 6.2 に表わしているようにハーフラインを跨ぐと試合を 200 サイクル経過するまでストップするようにしている．これは，相手チームの攻撃を考慮する必要がないため，無駄な動きをしないための設定を行った．

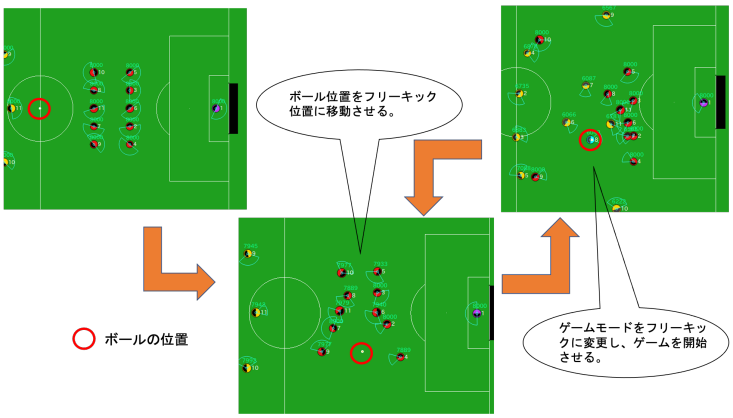


図 6.1: 試合の流れ

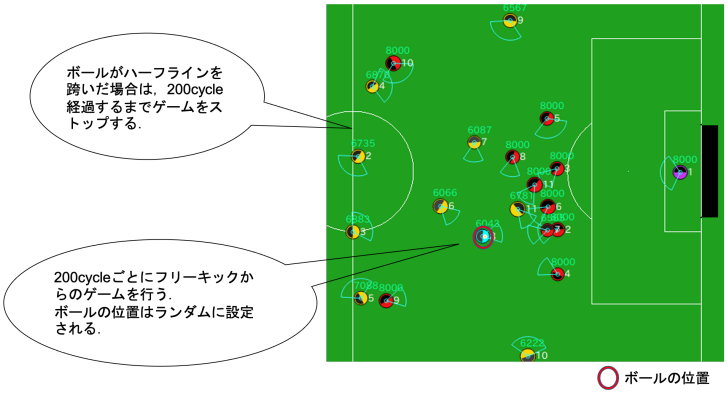


図 6.2: 試合条件

6.1.2. フィールド全体に適応するためのファジィ推論法

本節の実験でも 5 章と同様にファジィ推論法の一種である簡略型推論法を使用する．変更点としては，サッカーフィールド全体に適用するために前件部におけるメンバーシップ関数をサッカーフィールドにおける x 軸と y 軸の絶対値の 2 ルールで設定している．5 章と同様に取得する値はボールの位置座標である．サッカーフィールドの x 軸，y 軸に関しては図 4.2 を参考とする．x 軸，y 軸のメンバーシップ関数は，それぞれ図 6.3，図 6.4 に示す．図の通り x 軸は 6 ルール，y 軸は 5 ルールのメンバーシップ関数となるように設定している．

また，後件部は 5 × 6 の 30 ルールのシングルトンとしている．図 6.5 に概要を示す．図では 5 ルールのみの表示となっているが実際は $r_k^*(k = 1 \cdots 30)$ の 30 ルールとなっている．GA によって r_k^* の値をチューニングし，最適な値を探索する．図 6.6 にサッカーフィールドと前件部メンバーシップ関数の関係を示す．図の通りサッカーフィールドにおけるハーフラインから敵ゴール付近までを網羅するように設定している．ハーフラインから味方ゴール付近までをファジィ推論のルールに含めない理由としては，得点力を向上を目指すため攻撃の部分を重要視したいと考えているためである．

式 6.1 に今回設計したファジィルールを示す．R_5 までの表示となっているが，実際は R_{30} までの 30 のルール数となっている．また式 6.2 のように μ_k を求め，式 6.3 によって r を求める．最終的に式 6.4 によって評価値を求めるといった流れである．5 章と同様に

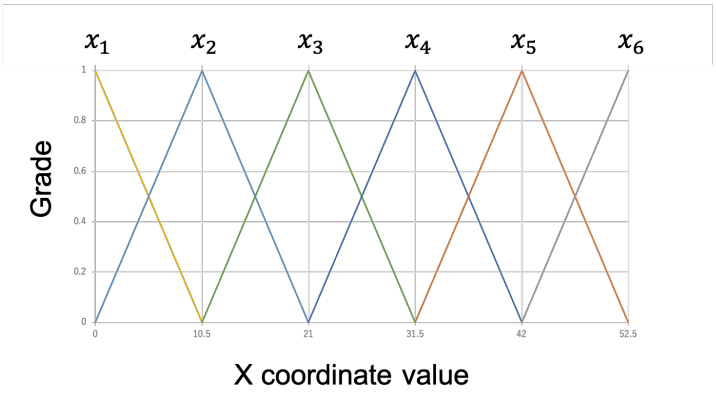


図 6.3: x 軸におけるメンバーシップ関数

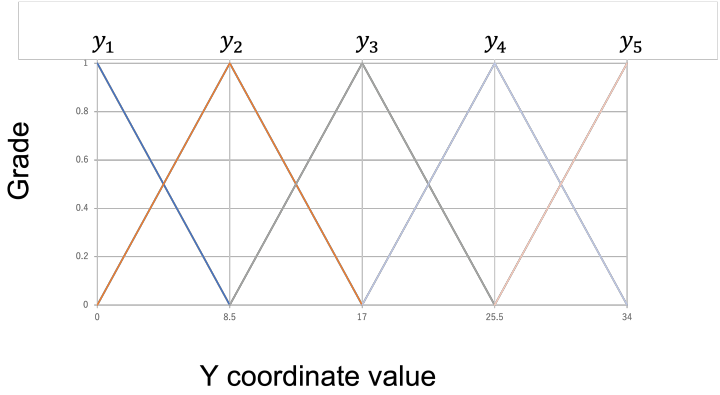


図 6.4: y 軸におけるメンバーシップ関数

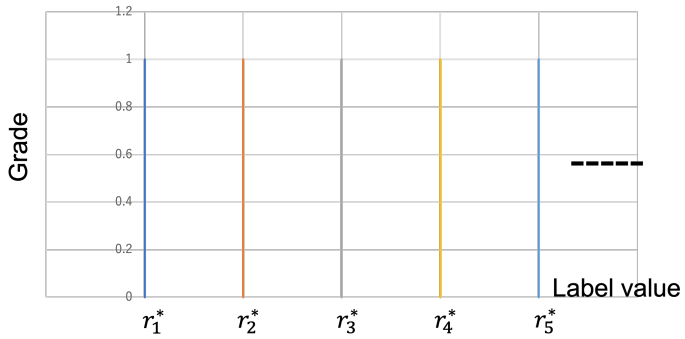


図 6.5: 後件部となるシングルトン

$U(x, y)$ は評価値である.

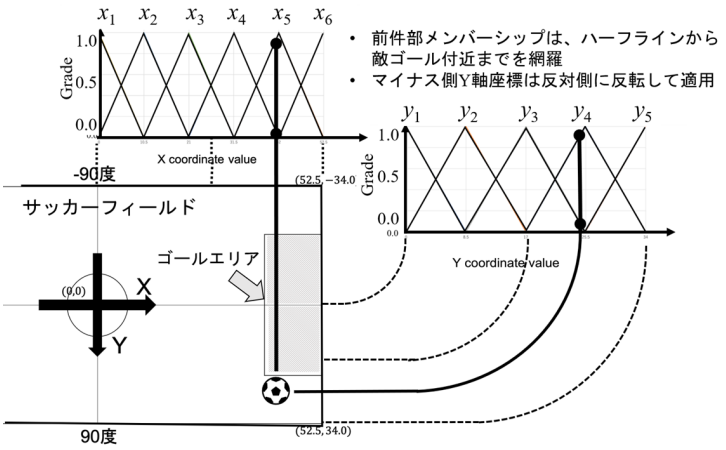


図 6.6: サッカーフィールドと前件部メンバーシップの関係

$$
\begin{array}{llllllllllll}
R_1 : If & x & is & x_1 & and & y & is & y_1 & Then & R & is & r_1^* \\
R_2 : If & x & is & x_2 & and & y & is & y_1 & Then & R & is & r_2^* \\
R_3 : If & x & is & x_3 & and & y & is & y_1 & Then & R & is & r_3^* \\
R_4 : If & x & is & x_4 & and & y & is & y_1 & Then & R & is & r_4^* \\
R_5 : If & x & is & x_5 & and & y & is & y_1 & Then & R & is & r_5^* \\
 & & & & & & \vdots & & & & & &
\end{array}
\tag{6.1}
$$

$$
\mu_k = \quad Grade \quad x_k(x) \quad \wedge \quad Grade \quad y_k(y) \tag{6.2}
$$

$$
r = \frac{\sum_{k=1}^{n} \mu_k r_k^*}{\sum_{k=1}^{n} \mu_k} \tag{6.3}
$$

$$
U(x, y) = x + r \tag{6.4}
$$

6.1.3. パラメータのチューニング

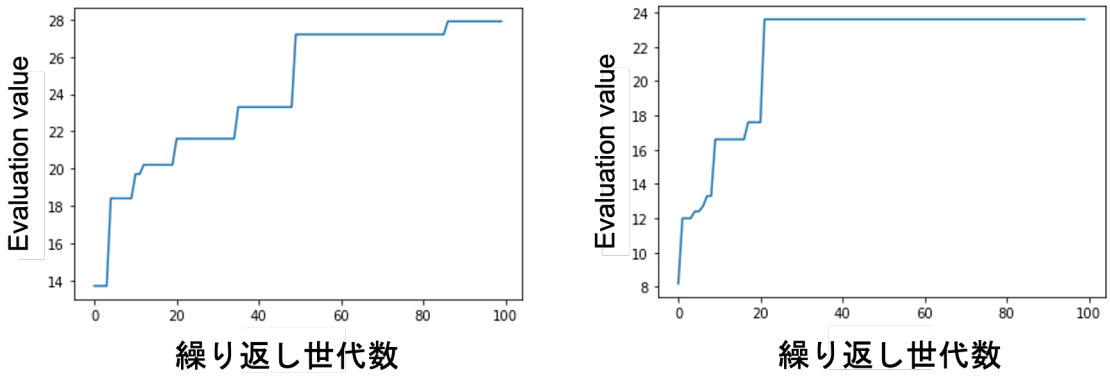
本実験では，チューニングするパラメータは r_k^* である．本実験でも同様に実数値 GA を使用する．実数値 GA の手順としては，図 5.5 の通りである．設定した実数値 GA のパラメータも 5 章と一部を除いて同様である．変更点としては，個体の次元数を 30 に，GA の集団内の個体数を 30 とした．また，本実験ではトーナメントサイズを 3 と 4 で同じ実験を行った．個体の次元数は探索範囲が 30 となったので 30 としている．個体数 30 とした理由としては，探索範囲を広げることによってよりよい評価関数を目指せると考えたためである．トーナメント数を 3，4 で実験を行った理由としては，トーナメント数が多い

ことによって局所的最適解に陥っている可能性があったので，今後の実験でも最適なトーナメント数を選択するために 3，4 で行い比較を行う．個体の評価を行う適合度関数としては式 6.5 を採用した．*point* は 1 試合における得点数，*shoot* は 1 試合におけるシュート数，*penalty* は 1 試合におけるペナルティエリアへの侵入数である．各指標に重み付けをしている理由としては，最も重要と考えている得点数に焦点を当て，得点能力を向上を目指していくためである．

$$fitness = 2point + shoot + 0.3penalty \tag{6.5}$$

6.1.4. 結果

実験を各トーナメント数ごとに行った．実験において各世代数における適合度の値の図を図 6.7 に示す．図 6.7a がトーナメント数 3 であり，図 6.7b がトーナメント数 4 の場合である．どちらの場合も世代が増すごとに適合度が上昇していることが見て取れる．また，トーナメントサイズ 3，4 の場合と既存の agant2d それぞれの対戦相手を既存の agent2d とした 100 試合の対戦成績を表 6.1 に示す．試合条件としては 6.1.1 項で述べた試合条件としている．サイズ 3 はトーナメントサイズ 3 の結果であり，サイズ 4 はトーナメントサイズ 4 の結果である．表から見て取れるように，トーナメントサイズ 3 の結果がチームの得点数において最も高い結果となった．シュート数の部分では，既存の agent2d が高い値であった．ペナルティエリアの侵入数の部分では，トーナメントサイズ 3 の結果が最も高い．トーナメントサイズ 4 の結果としては，すべての指標において下回っていた．



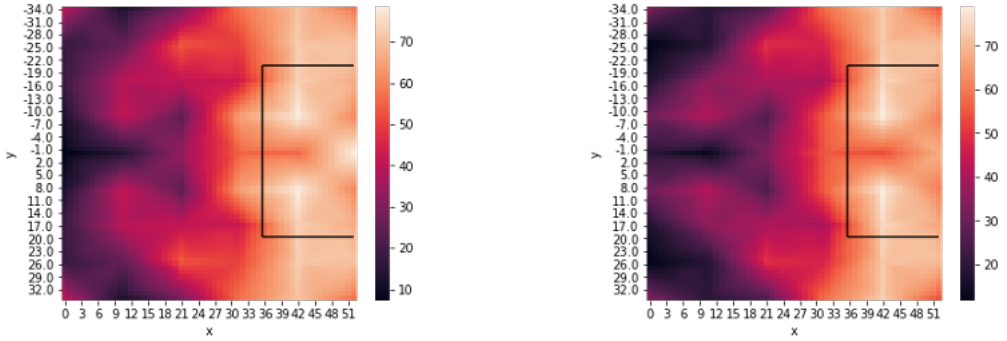
(a) サイズ 3　　(b) サイズ 4

図 6.7: 各トーナメントサイズに対するチューニング結果

図 6.8 はトーナメントサイズそれぞれにおける探索フレームワークの評価値を示す図である．6.8a がトーナメントサイズ 3 の場合であり，6.8b がトーナメントサイズ 4 の場合である．図から見て取れるようにそれぞれを比較したところ大きな違いは見て取れない．どちらもペナルティエリアの角の部分に最も評価値が高い部分が存在している．違いとしては，ゴール付近の評価値の値に違いが出ている．

表 6.1: 各チームによる 100 試合の対戦成績

	対戦評価					
	得点数 [点]		シュート数 [回]		ペナルティエリア侵入数 [回]	
各チーム	平均	±SD	平均	±SD	平均	±SD
サイズ 3	1.87	1.22	3.51	1.77	10.42	2.66
サイズ 4	1.36	1.20	2.86	1.63	8.58	2.57
agent2d	1.64	1.26	4.16	1.80	9.9	2.35



(a) サイズ 3　　(b) サイズ 4

図 6.8: 各トーナメントサイズに対する探索フレームワークの評価値

6.1.5. 考察

表 6.1 から見て取れるように，トーナメントサイズ 3，4 で結果に違いが現れた．やはりトーナメントサイズを大きくすることで局所的最適解に陥っている可能性がある．図 6.8 からも見て取れるように，トーナメントサイズ 2 つに大きな違いはほぼないがゴール付近の評価値の違いが結果に影響を与えていると考える．局所的最適解に陥らないためにもいくつかの条件で実験を行うことが重要であると考える．

また，本実験ではトーナメントサイズ 3 の結果が既存の agent2d を得点数の部分において上回った．4 章，5 章では，得点数の部分で上回ることができなかったので進歩である．評価関数をフィールド全体に適用したことで GA による探索範囲が広くなり，良い結果に繋がった．

6.2. 解探索範囲を拡張した評価関数の設計

6.2.1. 試合条件の変更点

6.1.1 項では，試合条件を変更して実験を行った．その試合条件に更に改良を加える．図 6.9 にその概要を示す．変更点としては，6.1.1 項ではハーフラインを跨ぐと一旦ゲームをストップしていたのだが，味方ペナルティエリア付近にボールがくるとゲームをストップするように改良した．理由としては，ハーフラインを跨ぐとゲームをストップしていたので，一旦自陣にボールを戻し攻撃を組み立てるという行為ができなかった．そのためランダムに設定されたボール位置がハーフライン付近となったときには攻撃ができずゲーム

がストップするという現象が起きていた．それを防ぐために試合条件を変更した．これによってより柔軟な攻撃ができると期待した．

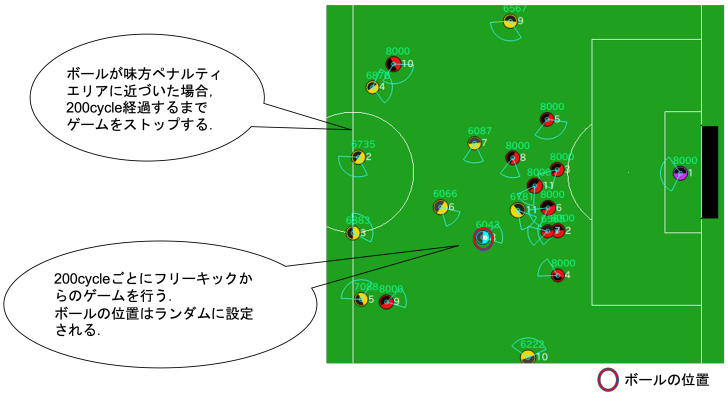


図 6.9: 試合条件

6.2.2. 解探索範囲をより広げたファジィ推論法

本節では 6.1 節で使用したファジィ推論法にさらに改良を加える．変更点としては，後件部のグレード値の上限を 1 にしたところを $\omega_k^*(k = 1 \cdots 30)$ とすることによって，GA による解探索をできるように変更を行った．図 6.10 にその概要を示す．また，式 6.3 にも変更点を加えた．式 6.6 に変更した式を示す．これにより GA の解探索範囲を広げてさらなる得点数の向上を目指した．

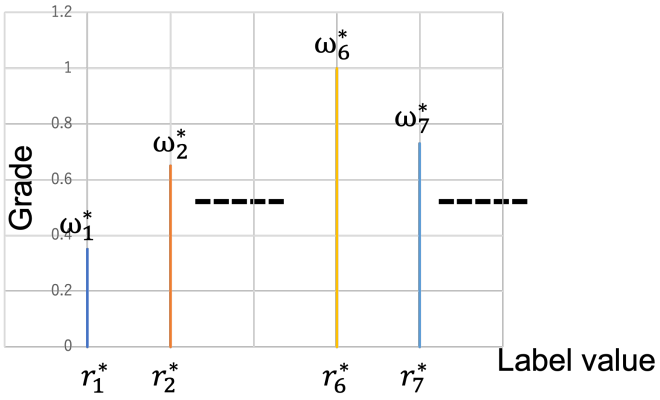


図 6.10: 後件部となるシングルトン

$$r = \frac{\sum_{k=1}^{n}(\omega_k^* \wedge \mu_k) r_k^*}{\sum_{k=1}^{n}(\omega_k^* \wedge \mu_k)} \tag{6.6}$$

6.2.3. パラメータのチューニング

本節でも実数値 GA を使用した．6.1 節との変更点としては，個体の次元数である．今回チューニングするパラメータとしては，r_k^* と ω_k^* であるため，個体の次元数は 60 として

いる．また，繰り返し世代数を 140 とした．理由としては，解探索範囲が広がったため，最適な解探索ができるまでより世代を要する可能性があると判断したため．また本節での実験ではトーナメントサイズ 3 でおこなっている．適合度関数は 6.1 節と同様の適合度関数を使用した．

6.2.4. 結果

実験を行った結果，繰り返し世代数とそれに対する適合度の推移は図 6.11 に示すようになった．図を見てわかることとしては，適合度は上昇しているが少ない世代の段階で上昇が止まっていることが見て取れる．しかし最終的な適合度は 5 章，6.1 節と比べて最も高くなった．また，表 6.2 に既存の agent2d との 100 試合の対戦成績の表を示す．試合条件としては，6.2.1 項で述べたものを採用している．比較対象として 6.1 節のトーナメントサイズ 3 の結果と既存の agent2d の結果を表示している．表から見て取れるように本節の結果の得点能力が最も高かった．シュート数では agent2d よりも下回っているがペナルティエリアの侵入数でも上位という結果となった．得点できる可能性の高いシュートを選択できていることが結果から分かる．

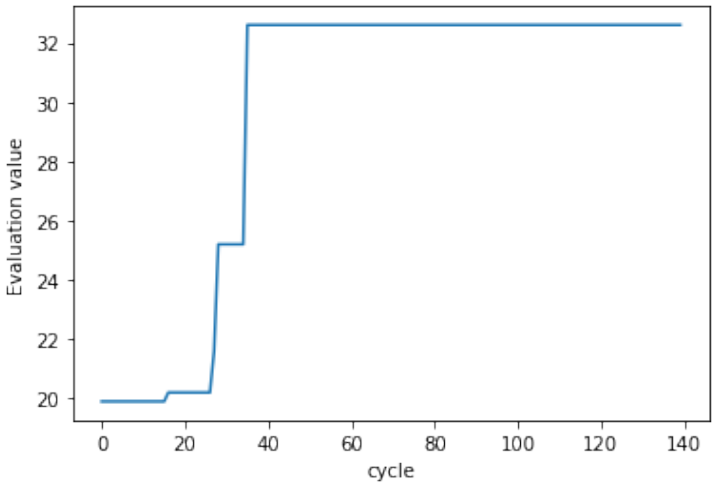


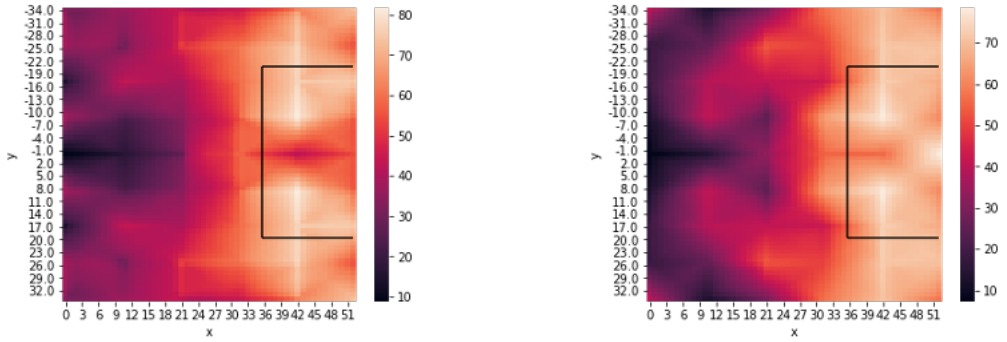
図 6.11: チューニング結果

表 6.2: 各チームによる 100 試合の対戦成績

	評価					
	チームの得点数 [点]		チームのシュート数 [回]		ペナ侵入数 [回]	
各チーム	平均	±SD	平均	±SD	平均	±SD
本節	2.1	±1.36	3.77	±1.99	10.91	±2.71
6.1 節	1.87	±1.22	3.51	±1.77	10.42	±2.66
agent2d	1.64	±1.26	4.16	±1.80	9.90	±2.35

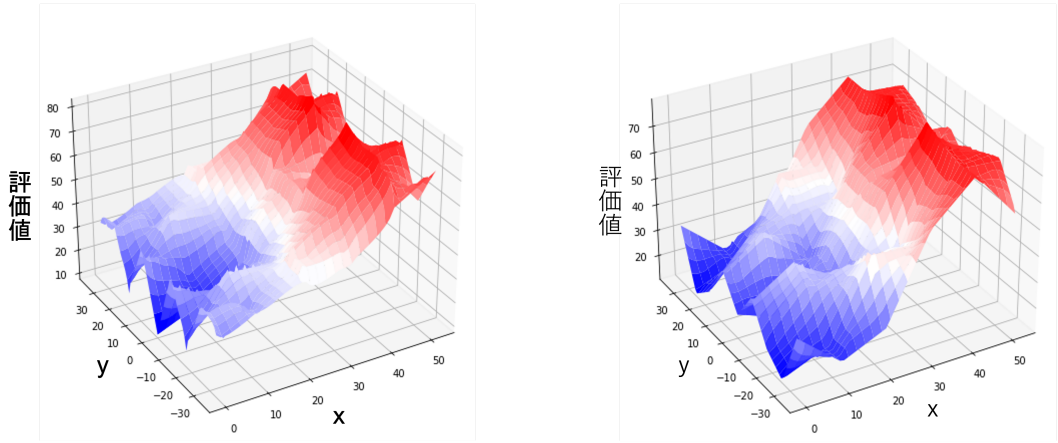
図 6.12 は本節と比較対象として 6.1 節のトーナメントサイズ 3 の場合での探索フレームワークにおける評価値である．図から見て取れることとしては，どちらもペナルティエリ

アの角の部分に評価値が高い部分が存在している．どちらの実験でもペナルティエリアの角の部分にボールを集めていることが分かる．違う部分としては，本節の実験結果の方がゴール付近の y 軸が 0 の近くの部分で低い評価値となっている点である．また，図 6.13 に 3D の場合での探索フレームワークにおける評価値を示す．図から見て取れることとしてはサイドの評価値の値がトーナメントサイズ 3 の場合ではほぼ一定の評価値となっているのに対し，本節での実験結果ではサイドエリアになるほど低くなっていることが見て取れる．



(a) 本節　　(b) 6.1 節のサイズ 3

図 6.12: 本節と 6.1 節における探索フレームワークの評価値



(a) 本節　　(b) 6.1 節のサイズ 3

図 6.13: 本節と 6.1 節における探索フレームワークの評価値 3D

6.2.5. 考察

本節では 6.1 節で行った実験に更に改良を加え実験を行った．その結果より得点能力の向上ができた．やはり GA の解探索範囲を広げることによってより最適な評価関数の探索に繋がることが考えられる．また，実験での試合条件を変更したことにより，一つのフリーキックからのスタートにおける点の取りこぼしが減少していることも考えられる．図 6.12 からはペナルティエリアの角の部分にボールを集めることが得点の期待が大きく

なるということが分かる．変わった点としては，ゴール近くの y 軸が 0 の付近の評価値が低くなっている．ボールを取られる危険性が高いということが見て取れる．また，ゴール付近からサイドエリアにかけて評価値が変化している．これは，サイドエリアからボールをペナルティエリアの角の部分に運ぶことによって得点の期待値が高くなるということが分かる．6.1 節でのトーナメントサイズ 3 の場合と比較しても評価値の高い部分と低い部分が顕著にあらわれている．

しかし，既存の agent2d と比較して大きく得点能力が向上しているということではない．本節では GA の探索範囲を広げて実験を行っているが，その探索範囲も 60 という少ない数である．これは入力値として用いているのがボールの位置座標という変数のみであるためである．今後の実験としては，ボールの位置座標の変数に加えて取得できる変数を用いて，得点能力の向上に期待できる変数の探索が必要になってくると考える．

6.3. 他チームに対する実験結果

6.1 節，6.2 節では，実験を行うにあたって対戦相手を既存の agent2d とした．それを他のチームとして実験を行った場合にどのような結果が得られるのか，他のチームに対して強さの向上，得点能力の向上を図ることができるのかを実験を通して数値化する．実験方法としては，既存の agent2d に対して最も高い得点能力を得ることができた 6.2 節の実験方法を用いて行う．

6.3.1. Jyo_sen2021 に対して

Jyo_sen とは秋葉原のプログラミングスクールのサッカー部によって結成されたチームである．中学生から社会人までの方が存在しており日々プログラミング能力の向上に励んでいる [22]．Jyo_sen は 2021 年に開催された RoboCup2d の大会である RoboCup2021 [23] で 16 チーム中 9 位につけており上位には劣るが確かな実力を誇るチームである．本項では，実験の対戦相手として RoboCup2021 に出場した Jyo_sen2021 のチームを使用する．

6.3.2. 結果

図 6.14 は各世代数に対する適合度を示した図である．図から世代数が増えるごとに適合度が上昇していることが見て取れる．しかし適合度の値としては，6.2 節と比較すると低い値となっている．表 6.3 に対戦相手を Jyo_sen2021 とした本節でのチューニング結果と既存の agent2d との 100 試合における対戦結果を示す．表から本節での実験結果と既存の agent2d に得点数において大きな変化がないことが見て取れる．シュート数では，既存の agent2d が上回っている．

図 6.15 に本節の実験結果における探索フレームワークの評価値の図を示す．また図 6.16 に評価値の 3D を示す．それらの図から見て取れることとして，高い評価値となっている部分としてはペナルティエリアの角の部分である．これは図 6.12a と比較しても大きな変化はない．変化している部分としては，ゴールの端の部分がやや高い評価値となっている．

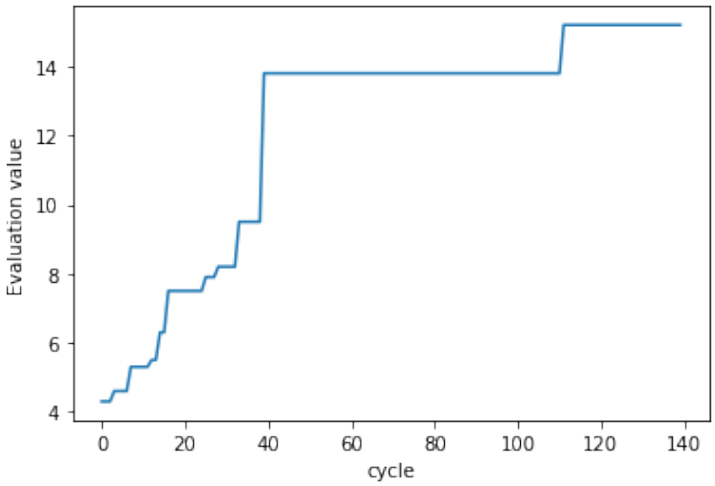


図 6.14: チューニング結果

表 6.3: 各チームによる 100 試合の対戦成績

	対戦評価					
	チームの得点数 [点]		チームのシュート数 [回]		ペナ侵入数 [回]	
各チーム	平均	±SD	平均	±SD	平均	±SD
実験結果	0.31	±0.52	0.67	±0.89	2.51	±1.94
agent2d	0.33	±0.60	1.03	±1.15	2.99	±1.98

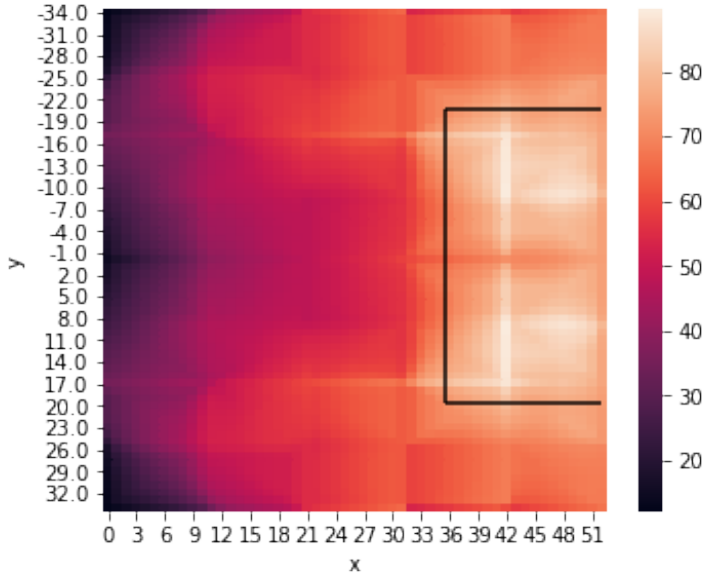


図 6.15: 本節における探索フレームワークの評価値

6.3.3. Persepolis に対して

Persepolis はもともと Razi という名前で 2012 年 11 月に学生によって活動をスタートし，IranOpne2018 で 3 位入賞するなどの成績を誇っている．2020 年に Persepolis という名前に改名された [24, 25]．RoboCup2021 では 16 チーム中 12 位の順位のチームである．

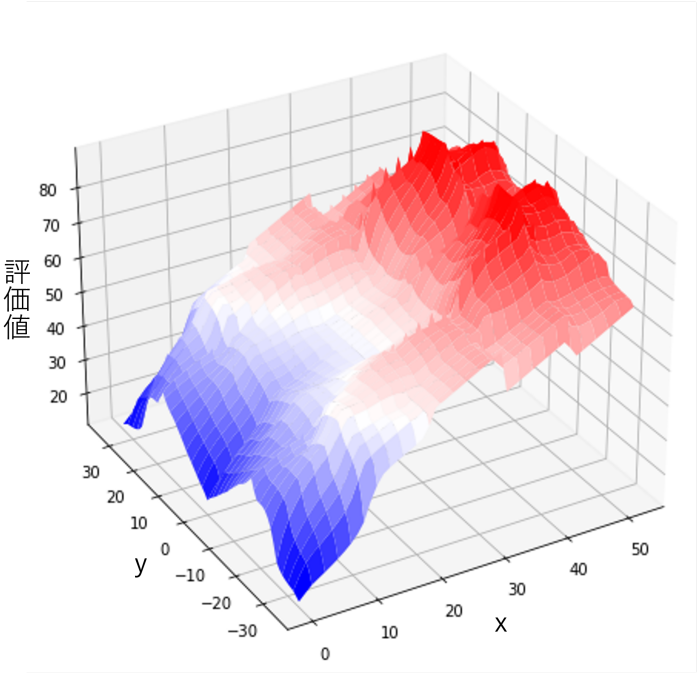


図 6.16: 本節における探索フレームワークの評価値 3D

本節では，RoboCup2021 で使用されたチームである Persepolis を対戦相手として実験を行う．

6.3.4. 結果

図 6.17 は各世代数に対する適合度を示した図である．図から世代数が増えるごとに適合度が上昇していることが見て取れる．しかし適合度の値としては，6.2 節と比較すると低い値となっている．表 6.4 に対戦相手を Persepolis とした本節でのチューニング結果と既存の agent2d との 100 試合における対戦結果を示す．表からチームの得点数において本節での実験結果がやや上回ってはいるが，得点能力に大きな変化はないことが分かる．シュート数では同程度のものとなった．ペナルティエリアへの侵入数では上回っていることが見て取れる．

表 6.4: 各チームによる 100 試合の対戦成績

	対戦評価					
	チームの得点数 [点]		チームのシュート数 [回]		ペナ侵入数 [回]	
各チーム	平均	±SD	平均	±SD	平均	±SD
実験結果	0.69	±0.71	1.31	±1.23	4.78	±2.22
agent2d	0.63	±0.82	1.31	±1.16	3.82	±1.73

図 6.18，図 6.19 に本節の実験結果における探索フレームワークの評価値を示す．図から見て取れることとしては，ゴール付近に高い評価値となる部分が存在しているということである．これまでの実験においては，基本的にはペナルティエリアの角の部分に高い評価値となる部分があったが Persepolis に対しては，違う結果となった．

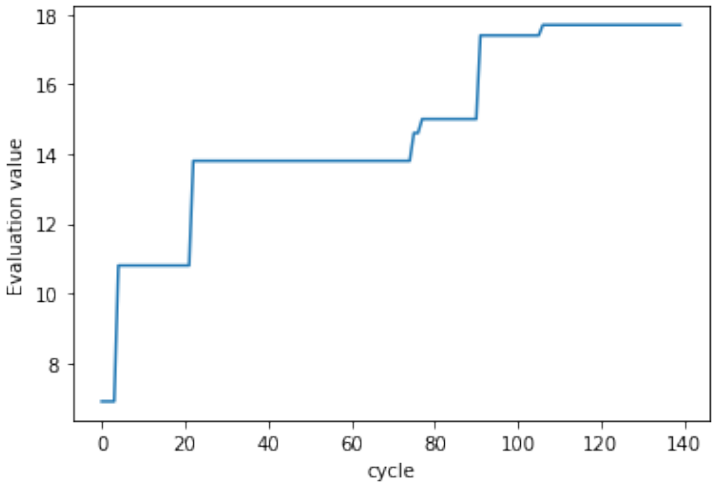


図 6.17: チューニング結果

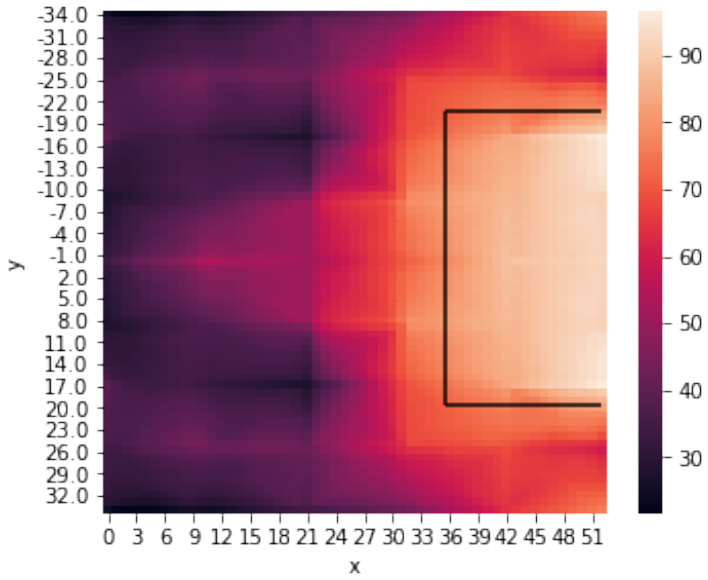


図 6.18: 本節における探索フレームワークの評価値

6.3.5. 考察

本節では対戦相手を Jyo_sen2021 と Persepolis として実験を行った．結果では，どちらのチームに対しても得点能力で既存の agent2d と同程度のものとなった．図 6.14，図 6.17 からも分かるようにチューニングによってチームの強さの引き上げはできているが，agent2d を上回るものには至らなかった．図 6.15 と 6.18 を比較すると高い評価値となっている部分に変化が起きている．図 6.15 では，フィールドにおいて評価値が低い部分や高い部分が特にない．最適な評価関数をチューニングしきれていないことが予測できる．図 6.18 ではフィールドにおいて低い評価値の部分が多くある．ボールを取られる可能性が高い部分を示すことができている．ゴール付近においては評価値に違いがあまりないためゴール付近における最適な評価値を探索しきれていない可能性がある．図 6.12 では，評価値の上下が明確になっているためゴールまでの最適なボールの運びかたが見て取れた．Jyo_sen2021 や Persepolis などの現役のチームに対しては，さらに改良を加え最適な

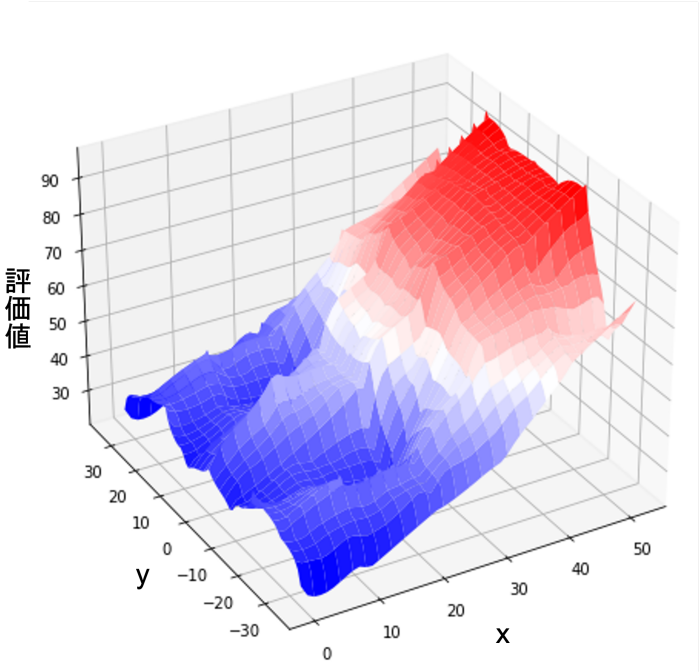


図 6.19: 本節における探索フレームワークの評価値 3D

評価関数に近づけていかなければならない．現実的に本節の実験の結果から各チームに対する戦術を構築していくには得点数やシュート数の数値として弱い部分ではあるが，今後より実験精度を高め，得点数やシュート数などの数値を向上できれば，実際の戦術構築の際に有効な手段となると考える．

第7章　結論

本論文では，サッカーのゲームの中で取得できるボールの位置座標から，得点数やシュート可能性が高まる位置を探索し，チームの得点能力の向上を目指すと共に，戦術構築の際に重要となるフィールド上で得点可能性の高い位置を示し，現実的なサッカーへ適用していくための第一歩となるという目的で研究を行った．はじめに，RoboCup2D の概要について述べ，RoboCup2D を始めるためのパッケージや agent2d についてまとめた．その後，チームの強さの指標を設定するために過去の大会のログを分析し，強さに関係する指標をまとめた．そして，評価関数の設計を主に 3 種類の方法で行った．一つはサッカーフィールド上の縦方向の位置座標である y 座標を用いて，決め打ちにより評価関数の設計を行った．2 つ目は，遺伝的アルゴリズムとファジィ推論法の一種である簡略型推論法を用いてサイド攻撃の向上を目指して評価関数を設計した．3 つ目は，2 つ目と同様に遺伝的アルゴリズムとファジィ推論法を用いてフィールド全体に適用できる評価関数を設計し，得点能力の向上を目指した．

まず，2020 年に開催された大会である夏の大会と JapanOpen2020 の試合データを LogAnalyzer3 を用いて解析し，自チームの得点数とそれ以外の評価指標との関係を見たところ，自チームのシュート数，相手チームのペナルティエリアへの侵入数が自チームが得点するために必要な評価指標だということがわかった．

次に，既存の agent2d の問題点であるサイド攻撃が単調という点を改善することによって，得点能力の向上が見られるのではないかという期待から，サッカーフィールド上の縦方向の位置座標である y 座標を用いて，y 座標の絶対値が 0 から離れるほど既存の評価関数に重みを与えることにより，相手ゴール周辺の評価関数の評価値の分布を変化させた．これよりサイド攻撃の幅は広がったが，チームの強さの実質的な向上には繋がらなかった．人間によって値を決め打ちして行う評価関数の設計は難しいと考えた．

そして，遺伝的アルゴリズムと簡略型推論法を用いた評価関数の設計を行った．設計した評価関数を適用しているのはサイド攻撃の場合のみである．実数値 GA によって簡略型推論法の後件部におけるシングルトンのラベル値をチューニングした．その結果，世代が進んで行くにつれて適応度を上昇させることに成功した．しかし，実質的にチームの強さを向上させることができなかった．考えられるのは，実際サイド攻撃が実質的な得点力の低下に陥っていないということ．それ以前のボールの運び方に問題があるということが考えられる．よって設計する評価関数の設定範囲をフィールド全体とし，ボールを集めるべき最適な箇所を探索することによって得点能力の向上が見られるのではないかと考えた．

3 つ目の評価関数として，フィールド全体に適用する評価関数を設計し，実数値 GA によってチューニングを行った．最終的な結果としては，既存の agent2d と比較して約 0.5 点程度の得点能力の向上が見られた．数値では大きな変化ではないが，得点能力が向上したということで一つの進歩である．また，実験のなかでの対戦相手を既存の agent2d としていたものを他のチームに変更し，実験によって他チームに対しても効果があるか検証を

行った．結果的には既存の agent2d と同程度のものとなった．他チームに対しては効果的とはならなかった．やはり GA による解探索範囲が小さいこと，入力値として扱っている指標がボールの位置座標のみという部分がより強いチームを相手とした場合に向上できなった理由ではないかと考えた．

最後に評価関数の設計とチーム作りに向けた今後の予定と研究課題・展望について述べる．本研究では，主にファジィ推論法と遺伝的アルゴリズムを用いて評価関数の設計を行った．既存の agent2d を対戦相手とした場合には得点能力を向上が見られた．また，得点可能性が高くなる位置を実験結果から視覚化することができた．これによって，対戦相手に特化した評価関数を構築することによって，戦術構築の際に有効になる可能性を示すことができた．しかし，より強いチームを対戦相手とした場合にはいい結果を得ることには至らなかった．考えられる点としては，上記でも述べたが，解探索範囲が小さいこと，入力値として扱っている指標がボールの位置座標のみということである．よって，サッカーの試合中に得られる指標を用いてさらに対戦相手に特化した評価関数を設計していく必要があると考える．例としては，ボール近くの敵の位置座標や，敵キーパーの位置である．それらとボールの位置座標を用いることによって，解探索範囲を広め，さらに有効な評価関数を設計を行うことができると考える．今後としては，上記に記載した点の改善を行っていく．そのためにも実験時間の短縮化を目指していかなければならない．本論文における実験には一度の実験を行うのに 3 日程度の時間がかかる．よって分散処理ができるシステムの構築が重要である．これによって，1 日程度から半日程度までの時間に効率化でき，実験の試行回数を増やすことができる．また，GA よりも効率的な手法の発見も実験時間の効率化に繋がる．GA では 1 世代のみの評価を行うために個体群の数の試合を行わなければならない．そのため何万回にもおける試合を実験のなかで行わなければならない．試合数を抑えより効率的に最適化していくことが実験時間の短縮に繋がる．実験時間の短縮を図りさらに実験を繰り返し行うことによってさらに得点能力の高いチームを作成していくことができる．そして，より対戦相手に特化した評価関数をチューニングによって設計することによって，実際のサッカーにおける戦術構築へもより有用なものとなっていく．今後も高みを目指して実験を行っていく．

謝辞

本研究を遂行するにあたり，多忙にもかかわらず数々のご指導・助言を賜りました高知工科大学システム工学群電子・光システム工学教室准教授星野孝総先生に心から深く感謝致します．また，RoboCup2d についての質問にいつもご丁寧にお答えしてくださった，岡山理科大学情報理工学部講師秋山英久先生，大阪府立大学大学院人間社会システム科学研究科教授中島智晴先生には，深く感謝致します．本システムの開発にあたり，高知工科大学システム工学群 Soft Intelligent System on a Chip 研究室の皆様には，日頃から様々な意見を頂きありがとうございます．最後になりましたが，大学・大学院生活 6 年間を支えてくれた両親・家族に深く感謝いたします．

参考文献

[1] Silver, David, Huang, Aja, Maddison, Chris J, Guez, Arthur, Sifre, Laurent, Van Den Driessche, George, Schrittwieser, Julian, Antonoglou, Ioannis, Panneershelvam, Veda, Lanctot, Marc, et al.: "Mastering the game of Go with deep neural networks and tree search", *nature*, Vol. 529, No. 7587, pp. 484–489, 2016.

[2] Silver, David, Schrittwieser, Julian, Simonyan, Karen, Antonoglou, Ioannis, Huang, Aja, Guez, Arthur, Hubert, Thomas, Baker, Lucas, Lai, Matthew, Bolton, Adrian, et al.: "Mastering the game of go without human knowledge", *nature*, Vol. 550, No. 7676, pp. 354–359, 2017.

[3] 秋山英久: " ロボカップサッカーシミュレーション 2D リーグ必勝ガイド", 秀和システム, 2006.

[4] "Prozone". `https://prozone-sports.ueniweb.com`. Accessed: 2022-1-10.

[5] "Tracab". `www.tracab.com`. Accessed: 2022-1-3.

[6] 秋山英久: " アクション連鎖探索によるオンライン戦術プランニング", 人工知能学会研究会資料, SIG-Challenge-B101-6, pp. 23–28, 2011.

[7] Yu, Junfeng, Bahitbek, Ardah, Yang, Yingze: "The Research of RoboCup2D Player Shooting Technique Based on Fuzzy Control", *Journal of Physics: Conference Series*, Vol. 2203, No. 1, p. 012059, 2022.

[8] Zare, Nader, Amini, Omid, Sayareh, Aref, Sarvmaili, Mahtab, Firouzkouhi, Arad, Rad, Saba Ramezani, Matwin, Stan, Soares, Amilcar: "Cyrus2D base: Source Code Base for RoboCup 2D Soccer Simulation League", *arXiv preprint arXiv:2211.08585*, 2022.

[9] 福島卓弥, 中島智晴, 秋山英久: "RoboCup サッカーにおける行動列からの評価関数モデリング", 日本知能情報ファジィ学会 ファジィ システム シンポジウム 講演論文集 第 34 回ファジィシステムシンポジウム, pp. 851–856. 日本知能情報ファジィ学会, 2018.

[10] 秋山英久: "RoboCup サッカー 2D シミュレーションリーグ解説: 仕組みと環境構築", 知能と情報, Vol. 23, No. 5, pp. 714–720, 2011.

[11] " ロボカップシミュレーションリーグ日本公式 Wiki". `http://rc-oz.osdn.jp/pukiwiki`. Accessed: 2020-12-18.

[12] Akiyama, Hidehisa, Nakashima, Tomoharu: ”Helios base: An open source package for the robocup soccer 2d simulation”, *Robot Soccer World Cup*, pp. 528–535. Springer, 2013.

[13] 秋山英久: ”RoboCup サッカー 2D シミュレーションリーグ解説: サンプルエージェントを使ったチーム開発”, 知能と情報, Vol. 23, No. 6, pp. 838–844, 2011.

[14] 山田剛史, 杉澤武俊, 村井潤一郎: ”R によるやさしい統計学”. 株式会社 オーム社, 2008.

[15] Holland, John Henry, et al.: ”Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence”. MIT press, 1992.

[16] 読売新聞大阪本社: ” 最速への挑戦—新幹線「N700 系」開発”. 東方出版, 2006.

[17] 高橋裕樹: ” 遺伝的アルゴリズム”, 映像情報メディア学会誌, Vol. 66, No. 3, pp. 209–211, 2012.

[18] 廣安知之, 三木光範, 福永隆宏, 福永隆宏: ” 遺伝的アルゴリズムにおける実数値ベクトル表現, 世代交代モデル, 母集団分割効果の検討”, 同志社大学理工学研究報告, Vol. 44, No. 1, pp. 25–35, 2003.

[19] 小林重信: ” 実数値 GA のフロンティア”, 人工知能学会論文誌, Vol. 24, No. 1, pp. 147–162, 2009.

[20] 星野孝総, 瀧本浩志: ” ライントレースカーの二段ファジィ制御による速度制御と差分進化アルゴリズムによる加速制御器の最適化”, 知能と情報, Vol. 25, No. 3, pp. 760–771, 2013.

[21] 菅野道夫: ” ファジィ制御”, 日刊工業新聞, 1991.

[22] Mochizuki, Minoru, Kataoka, Takashi, Ueda, Riku, Tatenuma, Ryoya, Sugihara, Ken, Nakamura, Masayoshi: ”Jyo_sen2021 (Japan) 2D Soccer Simulation League Team Description Paper”, *RoboCup 2021 Symposium and Competitions, Worldwide*, 2021.

[23] ”RoboCup2021”. `https://ssim.robocup.org/soccer-simulation-2d/2d-competition/2021-2/match-results/`. Accessed: 2022-1-5.

[24] Noohpisheh, Morteza, Shekarriz, Maryam, Bordbar, Afshin, Liaghat, Mohammadhassan, Salimi, Alireza, Borzoo, Daniel, Zarei, Alireza: ”Razi soccer 2D simulation team description paper 2019”, *RoboCup 2019 Symposium and Competitions: Team Description Papers. Sydney, Australia*, 2019.

[25] Noohpisheh, Morteza, Shekarriz, Maryam, Zaremehrjardi, Fatemeh, Khademi Ardekani, F, Khorsand, Seyedeh Afsoon: ”Persepolis soccer 2D simulation team description paper 2021”, *RoboCup 2021 Symposium and Competitions, Worldwide*, 2021.

[26] ”Opta Home”. http://www.optasports.com/. Accessed: 2020-12-13.

[27] 大堀杏, 福島卓弥, 中島智晴, 秋山英久: ”LogAnalyzer3: RoboCup サッカーシミュレーション 2D ログ解析ツール”, 日本知能情報ファジィ学会 ファジィ システム シンポジウム 講演論文集 第 34 回ファジィシステムシンポジウム, pp. 109–109. 日本知能情報ファジィ学会, 2018.

[28] 福島卓弥, 中島智晴, 秋山英久: ”LogAnalyzer3 を用いたサッカー戦略分析”, 日本知能情報ファジィ学会 ファジィ システム シンポジウム 講演論文集 第 35 回ファジィシステムシンポジウム, pp. 26–30. 日本知能情報ファジィ学会, 2019.

研究業績

1. 吉見奎吾, 星野孝総. ”RoboCup2D シミュレーションのフリーキック時におけるアクション連鎖探索の評価関数の設計と検証” 第 37 回ファジィシステムシンポジウム FSS2021 at Online. 2021.9. 口頭発表
   《FSS 優秀発表賞受賞》

2. Keigo Yoshimi and Yukinobu Hoshino. ”Design and validation of an evaluation function using the GA for the Fuzzy inference of the action chain search in RoboCup2D” The 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics at Online. 2021.11. 口頭発表

3. 吉見奎吾, 星野孝総. ”RoboCup2D シミュレーションのフリーキック時における アクション連鎖探索のファジィ推論法による設計と検証” 第 38 回ファジィシステムシンポジウム FSS2022 at Online. 2022.9. 口頭発表

4. Keigo Yoshimi and Yukinobu Hoshino. ”Design and Validation of Action Chain Search during Free Kick in RoboCup2D Simulation” 2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems at Online. 2022.11. 口頭発表

付 録A　開発を助けるパッケージ

2.3 章で解説した agent2d のように，RoboCup2D には開発を手助けするパッケージが用意されている．今回は実験で必要となったパッケージを取り上げる．

A.1.　librcsc

クライアントの開発を補助するために作られた汎用的なライブラリであり，GNU Autotools によるパッケージング，数学ライブラリ，プレイヤ，コーチ，トレーナのフレームワーク，ゲームログパーサライブラリなどが搭載されており，基本の行動は librcsc を使うことにより実装可能となる [11].

A.2.　soccerwindow2

soccerwindow2 はエージェントの開発補助を目的とした多機能ビューワプログラムである．rcssmonitor 互換のモニタクライアントとして動作するだけでなく，単体でログプレイヤとして動作可能である．タイムシフト再生機能が実装されており，試合実行中に巻き戻して試合を再生することもできる．さらに，プレイヤエージェントが出力するデバッグ情報を視覚的に表示するビジュアルデバッガとしても設計されている．

図 A.1 は soccerwindow2 が持つ機能の一つで，テキストによる更に詳細なデバッグメッセージ表示を実現している．エージェントが出力するデバッグメッセージには任意の文字列を使用でき，ログレベルによる表示内容の切り替えをサポートしている．図左のコマンドはログレベルのオプションを表しており，そのコマンドをオンにすることにより，それぞれのオプションが持つデバッグを表示するようになる．ほぼ無制限にデバッグ情報を表示できるため，デバッグに大量の情報出力が必要な場合に有効である [13].

A.3.　LogAnalyzer3

LogAnalyzer3 は Python3 環境で実行可能な RoboCup サッカーシミュレーション 2D リーグのログ解析ツールである．ゲームログを解析し，Opta [26] で採用されている指標を算出することができる．Opta とはサッカーの試合中のボールタッチを詳細に数値化する「プレイヤー・パフォーマンス分析データ」である．最新版の LogAnalyzer3 では，これらの指標算出に加えてキック分布など独自のゲーム分析機能が追加されている．LogAnalyzer3 を使って試合を分析するためには，サッカーサーバから出力された 2 種類のログファイル（rcg，rcl）が必要である．これらのログファイルは，試合を管理しているサッカーサーバが自動的に出力しているもので，各サイクルのプレイヤの位置，速度などあらゆる情報

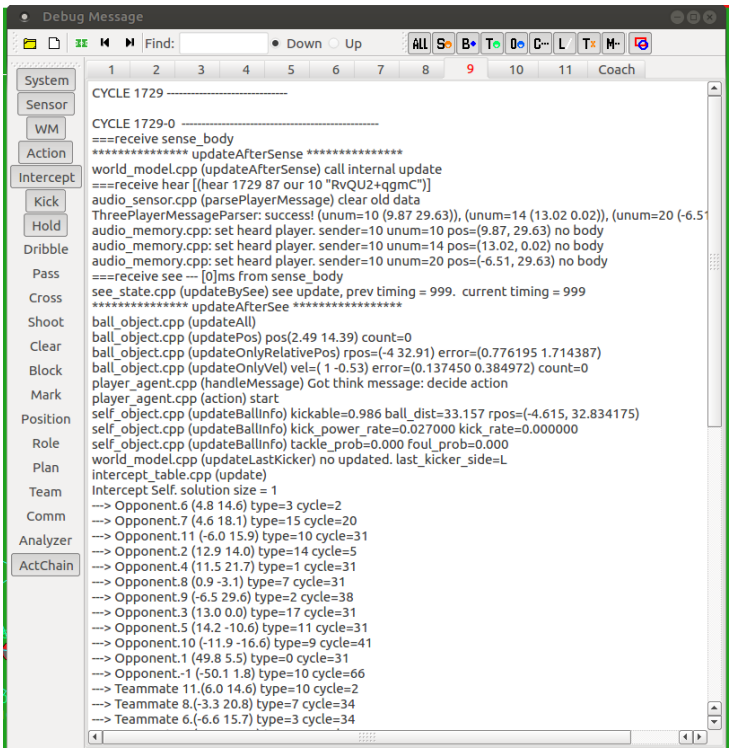


図 A.1: soccerwindow2 によるエージェントのデバッグメッセージ表示

が記録されている．ログファイルを利用することで終了した試合を再現することができる [27] [28].

付 録B　LogAnalyzer3が算出する評価指標

表 B.1: 評価指標と指標の意味

評価指標	指標の意味
data	日付
our_team	チーム名
our_final_team_point	自チームの得点
opp_final_team_point	敵チームの得点
our_penalty_shootout_point	自チームの PK 得点
opp_penalty_shootout_point	敵チームの PK 得点
final_result	最終結果
our_domination_time	自チームの支配時間
opp_domination_time	敵チームの支配時間
our_possession	自チームの支配率
opp_possession	敵チームの支配率
our_yellow	自チームのイエローカードの数
opp_yellow	敵チームのイエローカードの数
our_pass_all	自チームのパスの数
our_pass_L	自チームの左方向へのパス数
our_pass_R	自チームの右方向へのパス数
our_pass_F	自チームの正面方向へのパス数
our_pass_B	自チームの逆方向のみへのパス数
our_through_pass	自チームのスルーパス数
opp_through_pass	敵チームのスルーパス数
our_successed_tackle	自チームのタックル成功数
our_failed_tackle	自チームのタックル失敗数
opp_successed_tackle	敵チームのタックル成功数
opp_failed_tackle	敵チームのタックル失敗数
our_shoot	自チームのシュート数
opp_shoot	敵チームのシュート数
our_point	自チームの得点数
opp_point	敵チームの得点数
our_dribble	自チームのドリブル数
opp_dribble	敵チームのドリブル数
our_penalty_area	敵チームのペナルティエリアへの侵入数
opp_penalty_area	自チームのペナルティエリアへの侵入数
our_disconnected_player	自チームの切断数
opp_disconnected_player	敵チームの切断数